



# openLookeng ClickHouse Connector 在多数据中心下的应用探索

何正杰

信工所MESA团队研发工程师

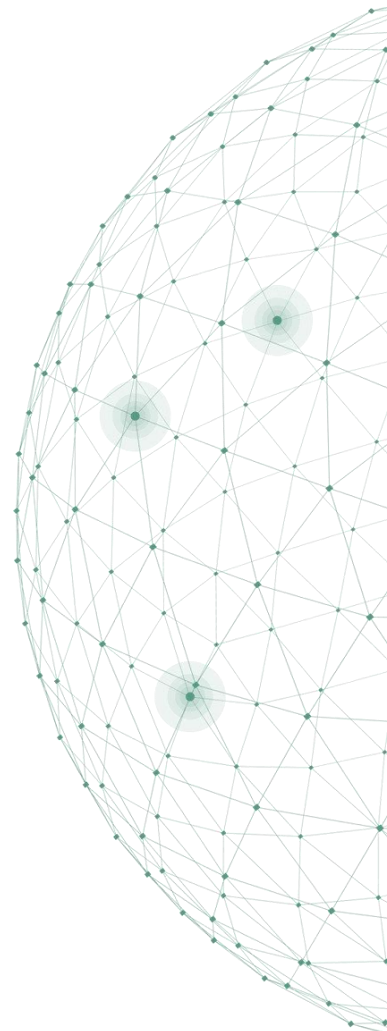
# 目录

- 01 信息工程研究所 MESA 团队介绍
- 02 Galaxy 网络流量数据平台
- 03 openLookeng 在 Galaxy 数据虚拟化引擎架构的应用实践
- 04 openLookeng ClickHouse connector 开发

# 01

## 信息工程研究所 MESA 团队介绍

- ▲ 科研方向
- ▲ 成果展示



# 中国科学院信息工程研究所

- 研究所是**中科院网络空间安全领域的总体所**，面向国家战略需求，在信息安全科技领域，开展基础理论与前沿技术研究，开发应用性技术与系统，**为国家信息化进程提供核心关键技术支撑与系统解决方案**。

- 攻研信息安全关键技术的**核心单位**

- 统筹信息安全重大项目的**总体单位**

- 引领安全科技发展方向**的龙头单位**

张弛有度 开合有法 矛盾兼容 软硬兼修

- 拥有涉密计算机系统集成资质认证甲级资质

- 通过ISO9000质量管理体系认证

定位明确

资质规范

# 中科院信工所 MESA 团队介绍

- 面向网络空间安全，研究大规模网络智能信息处理的基本理论、模型、算法和关键技术，研制可扩展、高可用、易使用的网络数据处理与分析系统
- 主要研究方向：大规模高性能网络信息处理

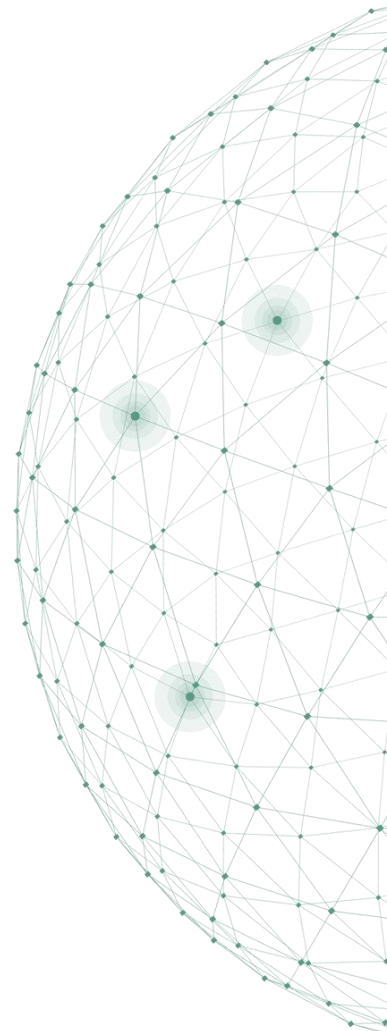


0

2

## Galaxy 网络流量数据平台

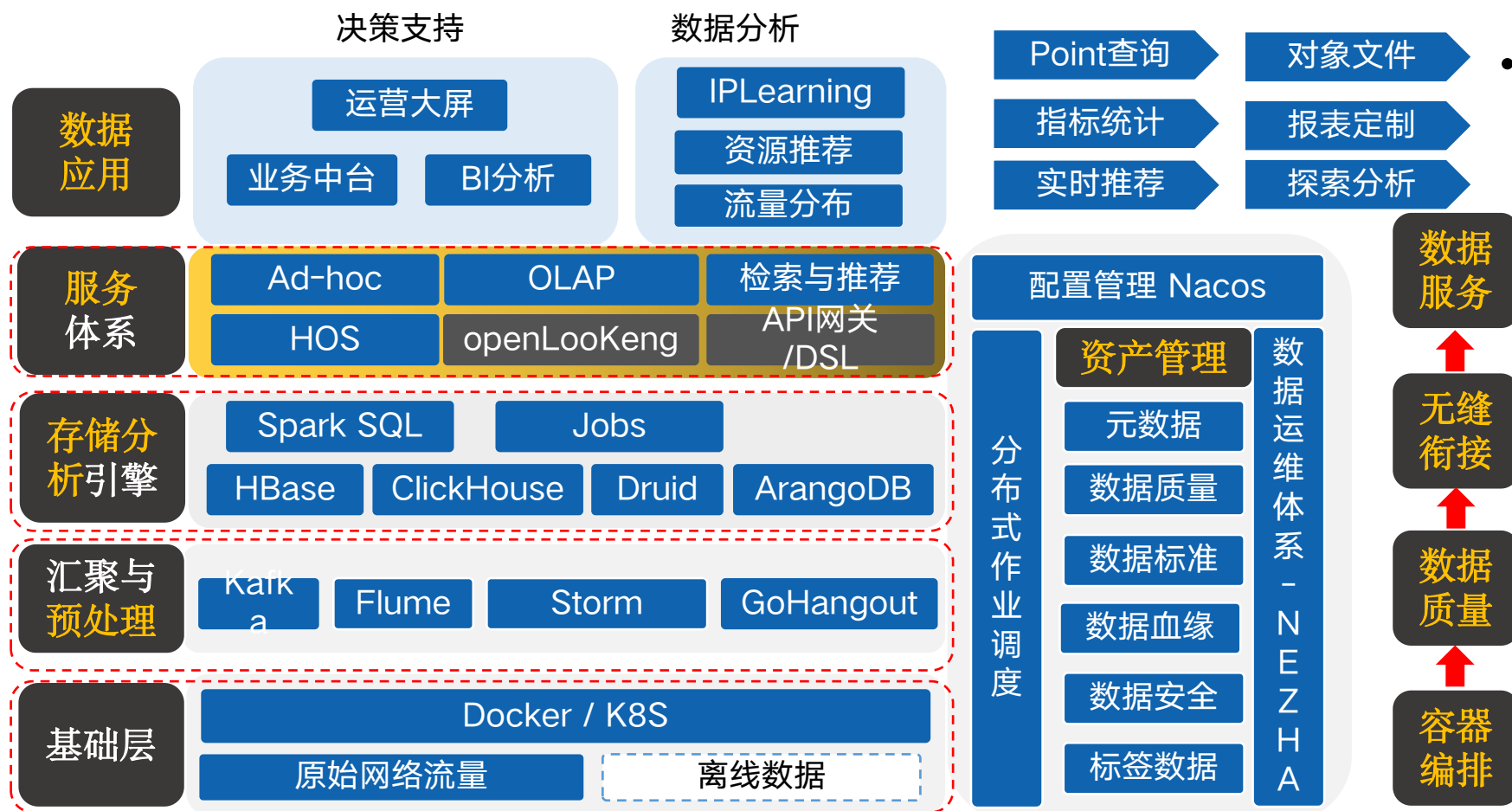
- ▲ 组件和作用
- ▲ 功能以及应用场景





# Galaxy 网络流量数据平台

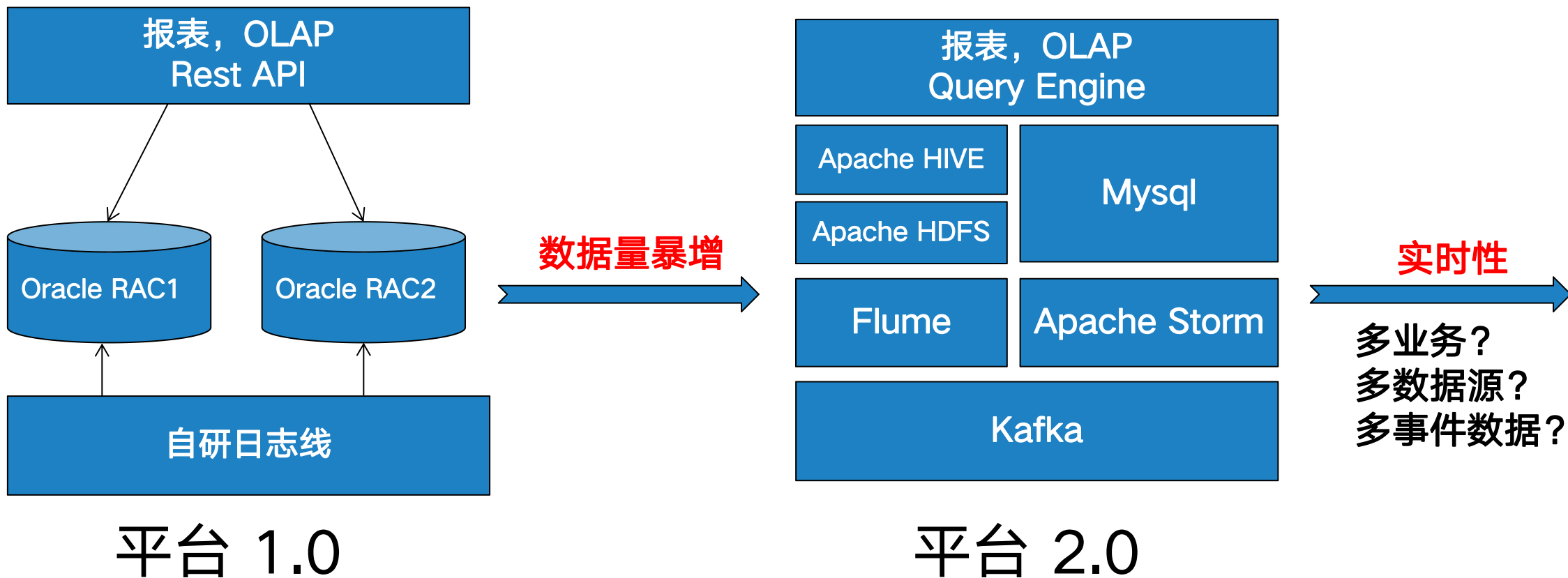
Galaxy 网络流量数据平台是一个网络流量存储、检索和探索分析平台。集成大数据开源软件技术，构建统一的数据服务，定制标准的元数据，目标帮助开发和分析人员提供统一接入口径，从数据中获取核心的业务价值。



## • 通用的流量日志处理方案

- 构建数据虚拟化引擎
- 实时数仓能力
- 简单大数据处理方案
- 构建统一数据资产

# 技术演进

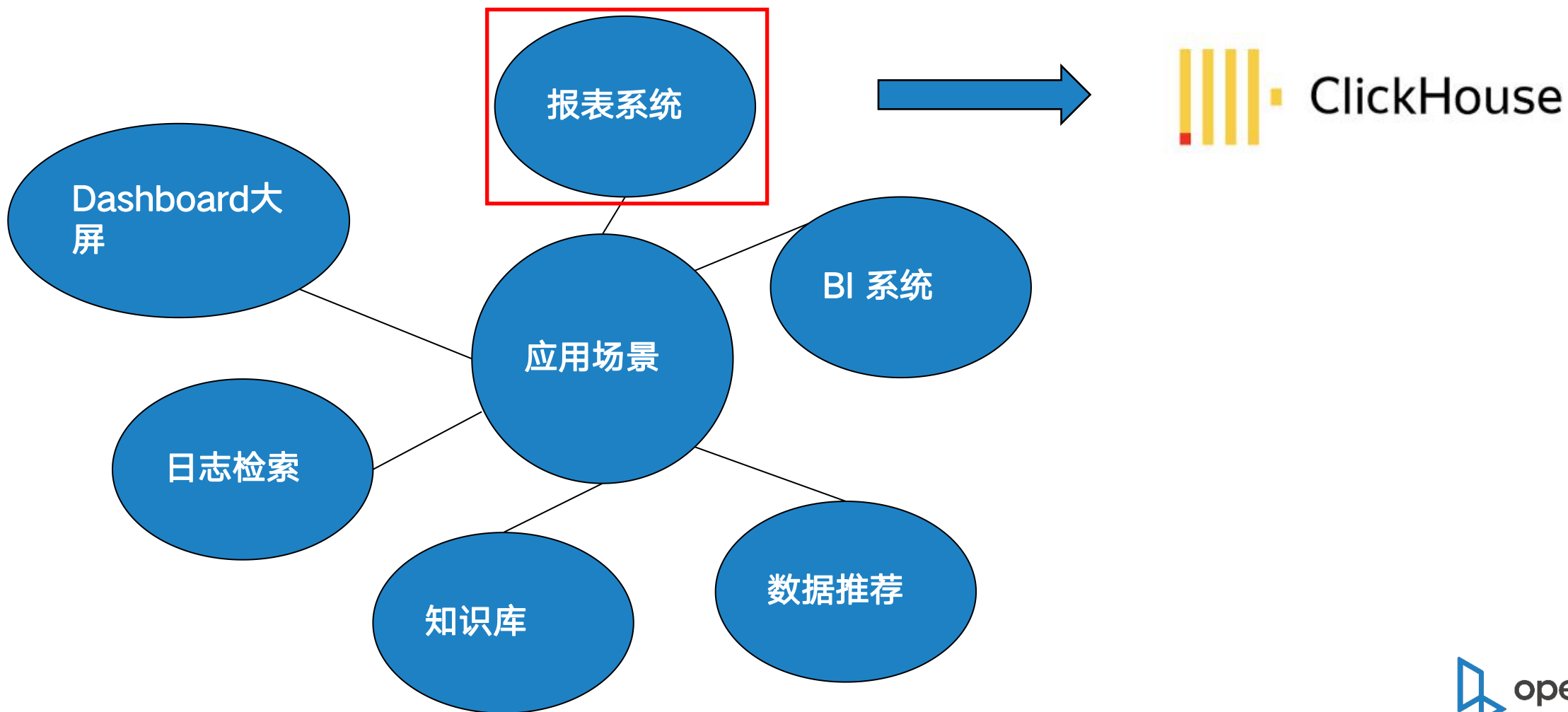




# | Galaxy 网络流量数据平台

场景	特点	技术
指标统计、实时推荐	实效性 近似计算	列存快速分析、时序、 存算一体
OLAP	准实时 多维数据探索分析	列存快速分析
网络实体画像 关系检索	亿级实体 十亿级关系实时检索更新	图模型
报表定制	任意维度 PB级	实时数仓
对象存储	海量数据存储	LSM、MOB

# Galaxy 网络流量数据平台

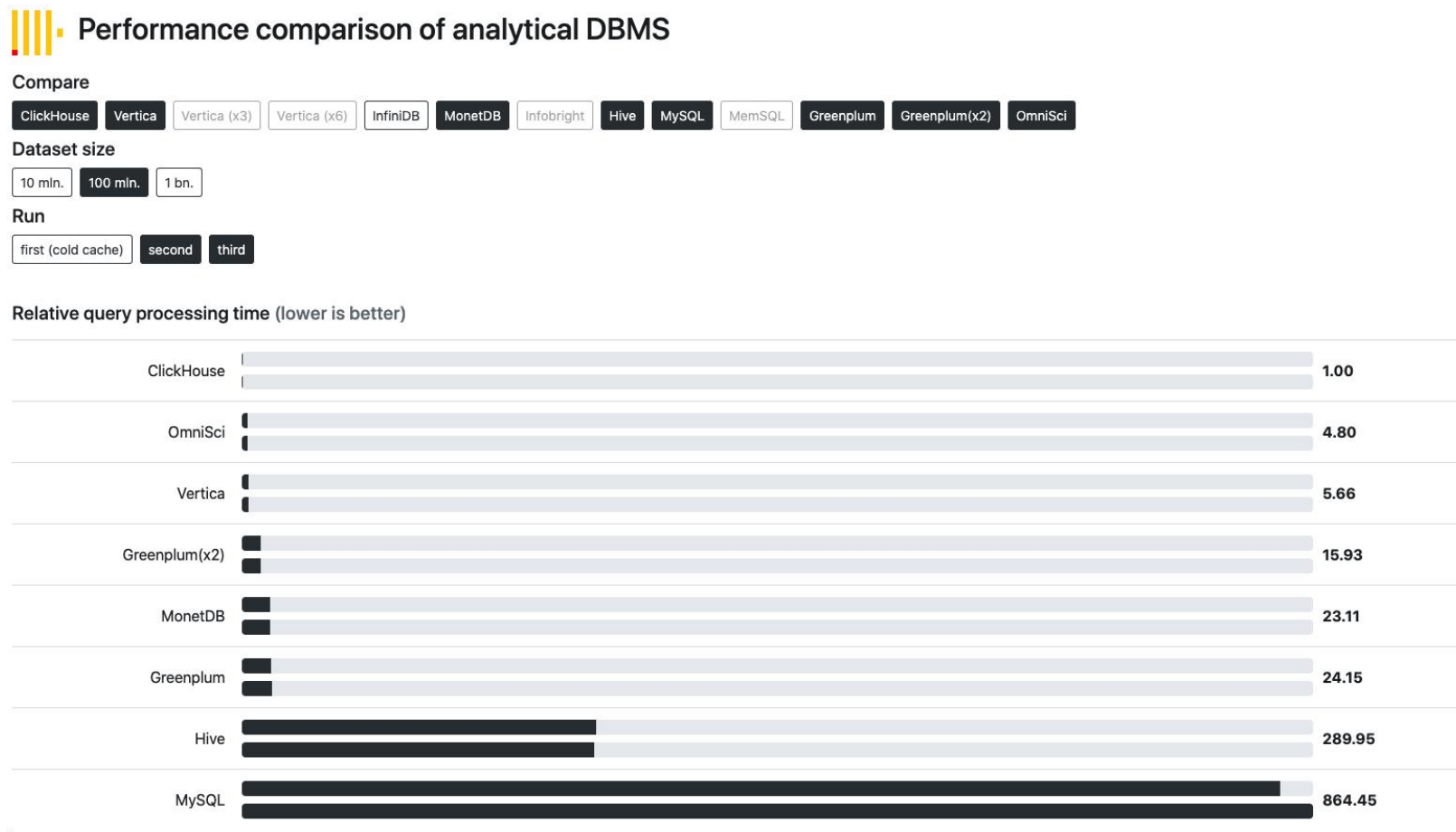


# | ClickHouse 简介

- ClickHouse是Yandex开源的一个用于联机分析(OLAP)的列式数据库管理系统(DBMS)
- 在多个测试结果中，ClickHouse表现出了比同类可比较产品更优的性能
- 注意点
  - 不支持事务
  - 对于更新和删除支持较低
  - 多表Join的性能不如单表查询
  - 非完全兼容ANSI SQL标准
  - 高并发支持有限，官方建议qps为100

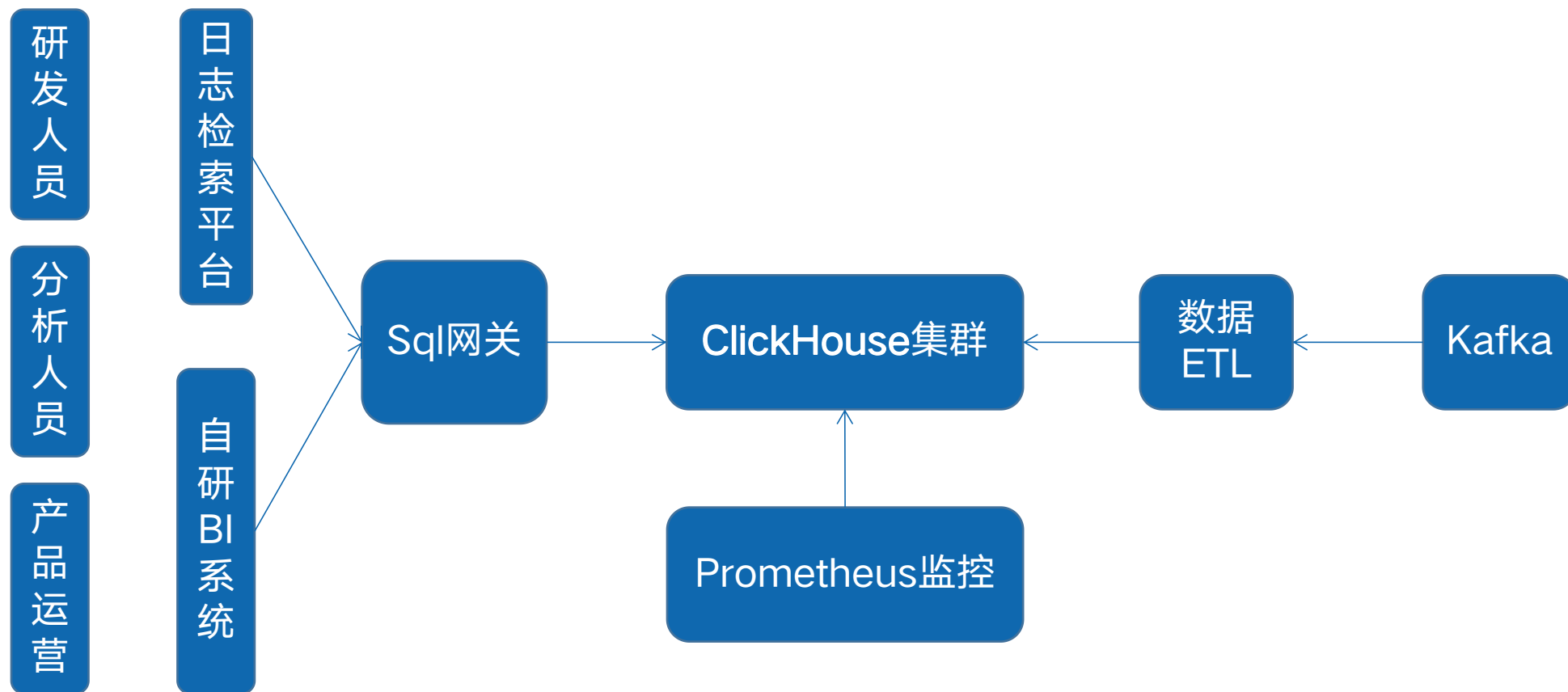
Ref: <https://clickhouse.tech/#independent-benchmarks>

# ClickHouse 简介



Ref: <https://clickhouse.tech/benchmark/dbms>

# | openLookKeng 在 Galaxy 中的应用实践

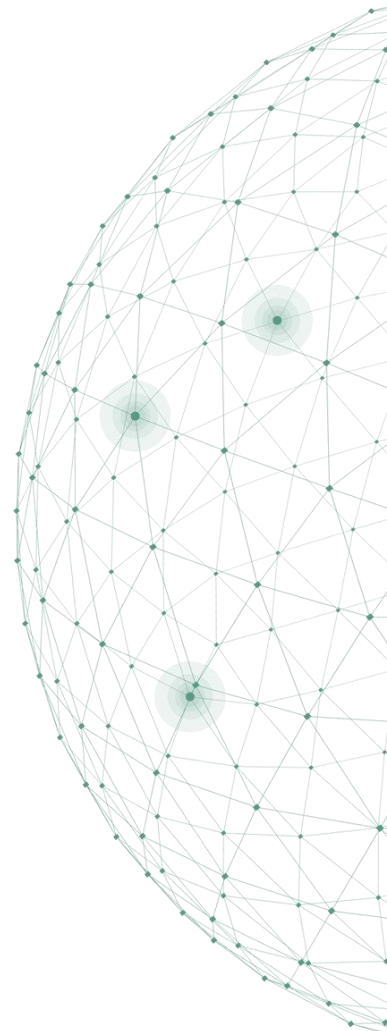


0

3

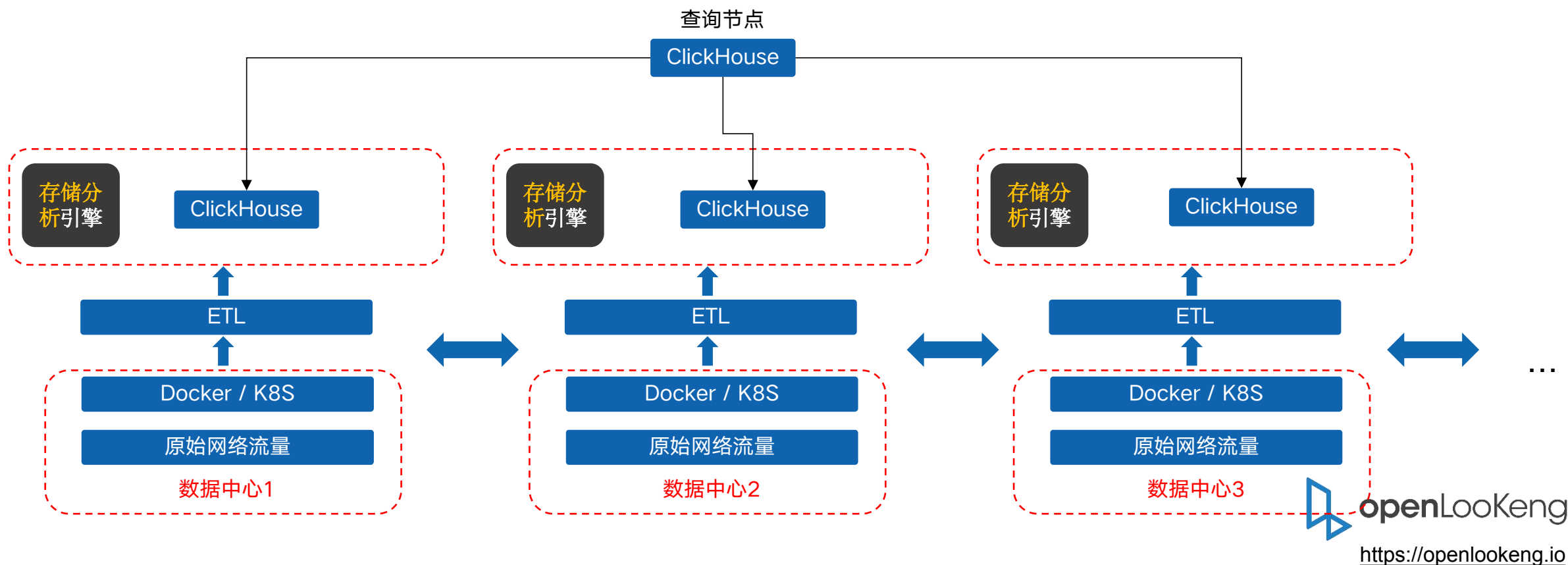
## openLookEng 在 Galaxy 中的应用实践

- ▲ 数据引擎面临的难点痛点
- ▲ 在哪一层接入openLookEng，为什么需要它



# | openLookEng 在 Galaxy 中的应用实践

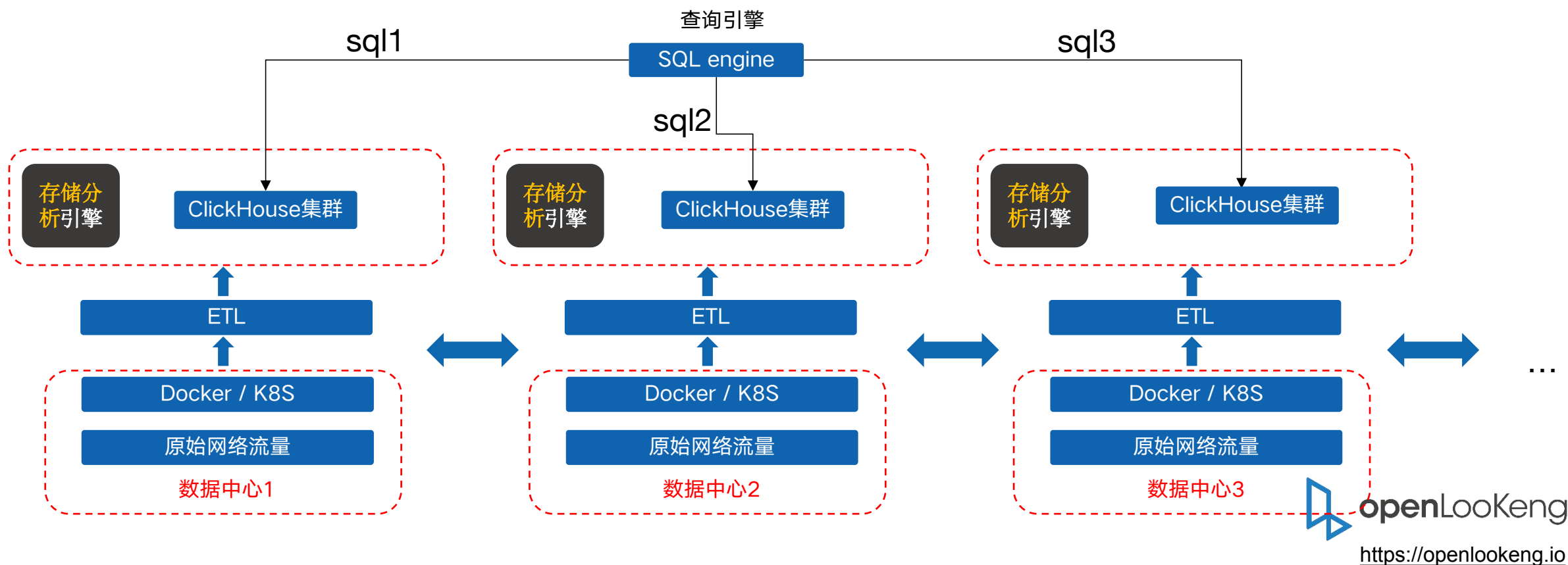
- 数据引擎面临的难点痛点
- 方案一：跨数据中心部署ClickHouse



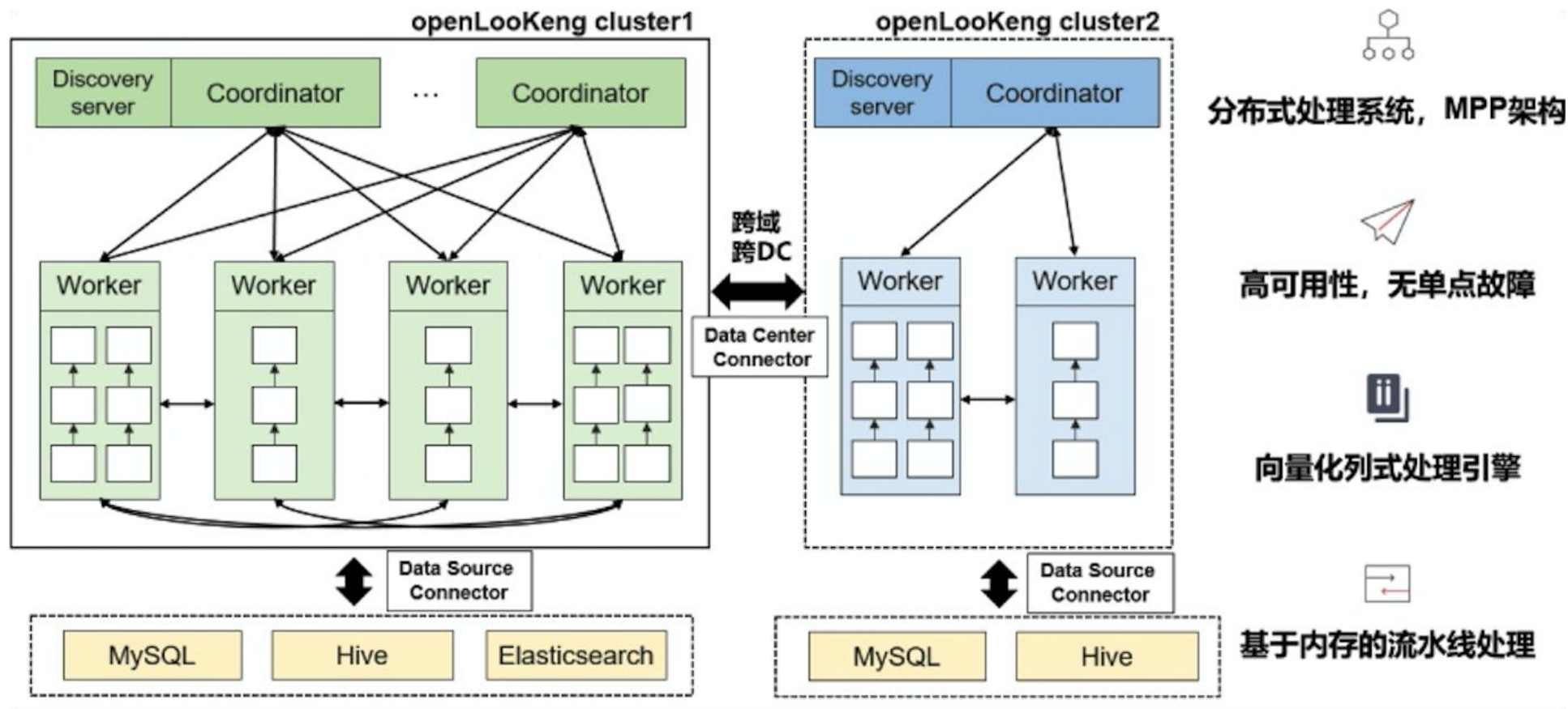


# openLookKeng 在 Galaxy 中的应用实践

- 数据引擎面临的难点痛点
- 方案二：查询引擎对拆分sql数据再同一汇聚



# openLookKeng 给 Galaxy 平台带来更灵活的方案选择

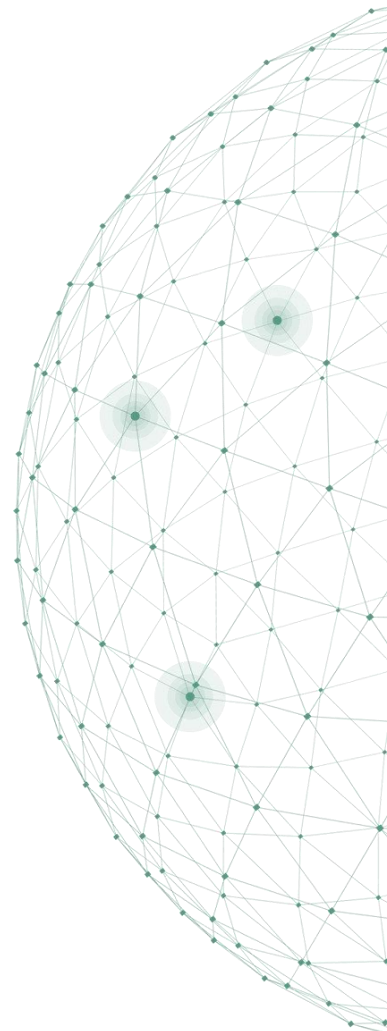


Ref: 罗旦 华为高级开发工程师

# 0/4

## ClickHouse connector 开发分享

- ▲ CH connector 实现的功能
- ▲ CH connector 1.1和1.2的测试



# ClickHouse connector 实现

- 数据类型转换：ClickHouse -> openLooKeng

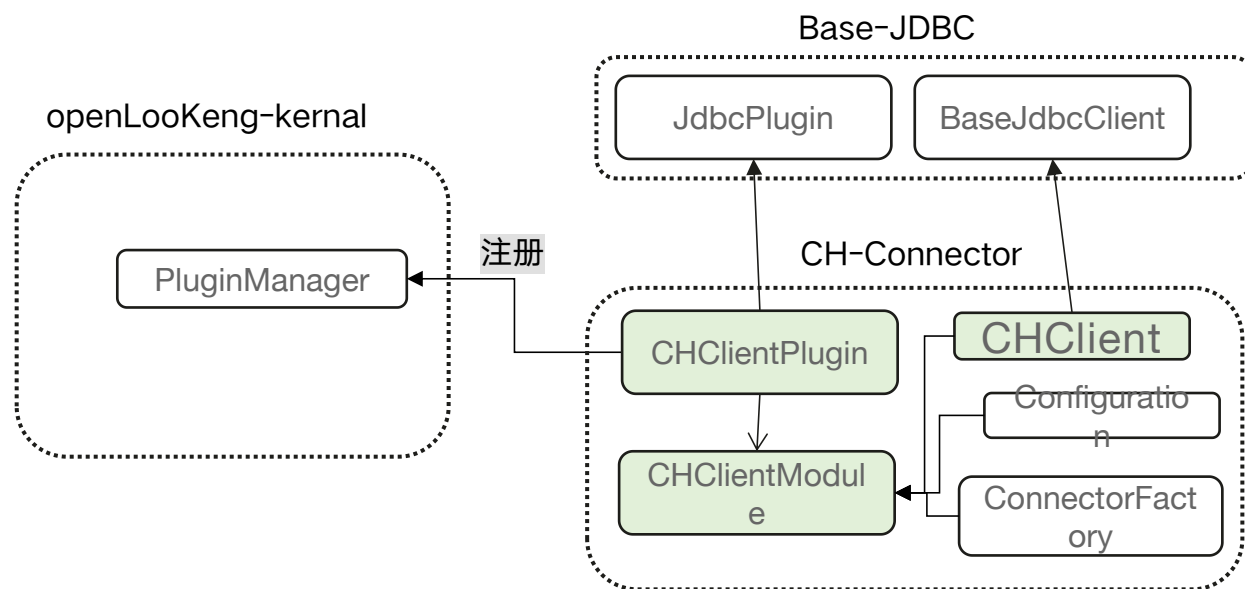
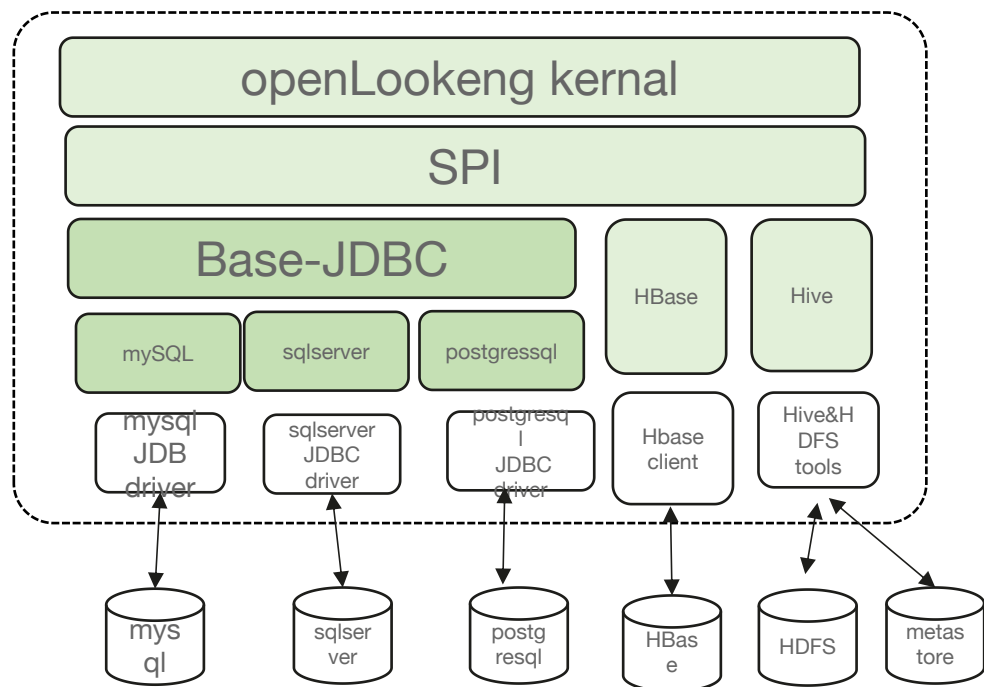
ClickHouse类型	openLooKeng类型	说明
UUID	UUID	
Int8	TINYINT	
Int16	SMALLINT	
Int32	INTEGER	
Int64	BIGINT	?
float32	REAL	?
float64	DOUBLE	?
DECIMAL(P,S)	DECIMAL(P,S)	
DECIMAL32(S)	DECIMAL(P,S)	无需处理
DECIMAL64(S)	DECIMAL(P,S)	
DECIMAL128(S)	DECIMAL(P,S)	
String	varchar	
DateTime	TIMESTAMP	
DateTime64	?	
Fixedstring(N)	char	
Enum	?	
UInt8	Int16	无需处理，区分大小写
UInt16	Int32	
UInt32	Int64	
UInt64	?	
Int128,Int256,UInt256	?	1111

openLooKeng类型	ClickHouse数据库类型	说明
BOOLEAN	Int8	
UUID	UUID	
TINYINT	Int8	
SMALLINT	Int16	
INTEGER	Int32	ck可以自动转换
BIGINT	Int64	
REAL	float32	ck可以自动转换
DOUBLE	float64	
DECIMAL(P,S)	DECIMAL(P,S)	
IPADDRESS	根据v4或者v6分别转换为IPv4/IPv6	待测试
varchar	String	
varchar(n)	String	待测试
CHAR(n)	FixedString(n)	
VARBINARY	String	
JSON	无	
DATE	DATE	
TIME	无	
TIME WITH TIME ZONE	无	
TIMESTAMP	DateTime	
TIMESTAMP WITH TIME ZONE	无	

Code:  
[https://github.com/Heatao/openLooKeng\\_ClickHouseConnector](https://github.com/Heatao/openLooKeng_ClickHouseConnector)

# ClickHouse connector 适配开发

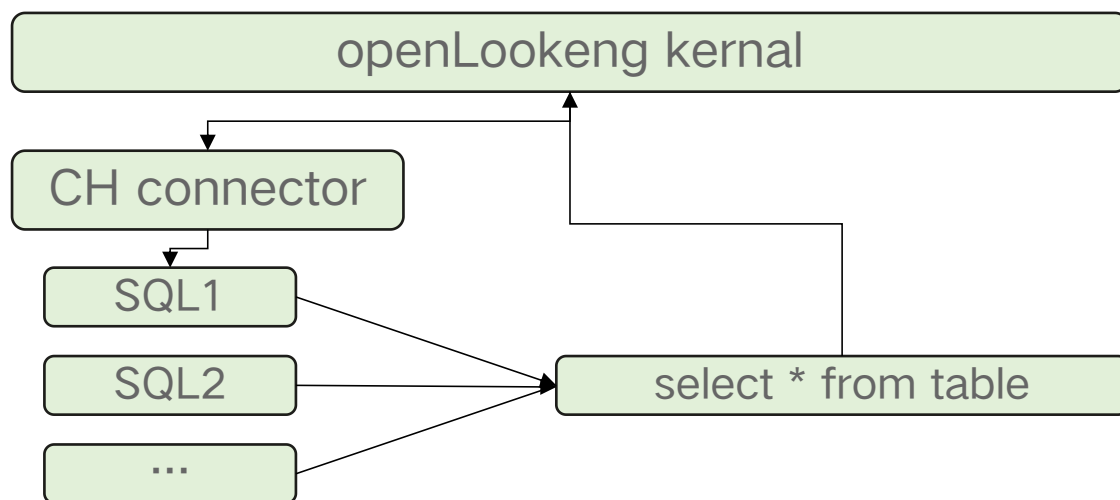
- 基于Base JDBC开发CH connector



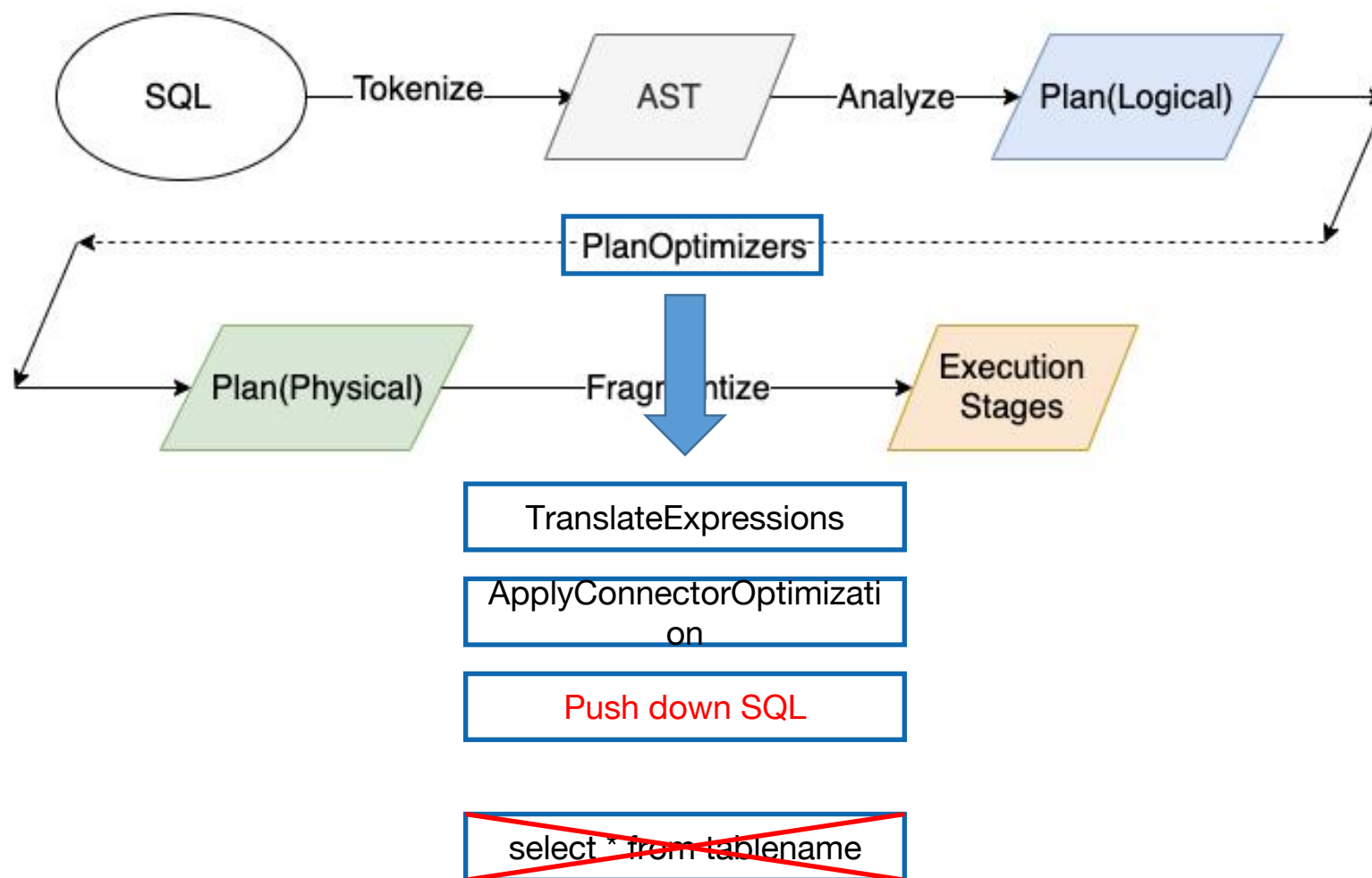
Ref: 华为高级工程师黎一泽, 王明月的技术分享 <https://space.bilibili.com/627629884/video>

# ClickHouse connector 适配开发

- 非下推情况
  - io.prestosql.plugin.jdbc.JdbcPlugin
  - io.prestosql.plugin.jdbc.JdbcClient
  - io.prestosql.plugin.jdbc.JdbcModule
  - io.prestosql.plugin.jdbc.BaseJdbcConfig
- 在openLookEng kernel计算
- 无法利用数据源的优势



# | openLookEng 下推





# ClickHouse connector 实现

- 下推算子
  - 继承Base JDBC connector
  - ClickHouse非标准SQL
    - 对于表达式的微调整
    - 对函数的重写

```
@Override
public String aggregation(List<Selection> symbols, Optional<List<String>> groupingKeysOp, Optional<String> groupIdElementOp, String from)
{
    StringBuilder builder = new StringBuilder();
    builder.append(from);
    if (groupingKeysOp.isPresent()) {
        List<String> groupingKeys = groupingKeysOp.get();
        if (!groupingKeys.isEmpty()) {
            builder.append(" GROUP BY ");
            builder.append(Joiner.on(", ").join(groupingKeys));
        }
    }
    else if (groupIdElementOp.isPresent()) {
        String groupEleStr = groupIdElementOp.get();
        builder.append(" GROUP BY GROUPING SETS ");
        builder.append(groupEleStr);
    }
    return select(symbols, builder.toString());
}
```

```
@Override
public String aggregation(List<Selection> symbols, Optional<List<String>> groupingKeysOp, Optional<String> groupIdElementOp, String from)
{
    StringBuilder builder = new StringBuilder();
    builder.append(from);
    if (groupingKeysOp.isPresent()) {
        List<String> groupingKeys = groupingKeysOp.get();
        if (!groupingKeys.isEmpty()) {
            builder.append(" GROUP BY ");
            builder.append(Joiner.on(", ").join(groupingKeys));
        }
    }
    else if (groupIdElementOp.isPresent()) {
        throw new UnsupportedOperationException("ClickHouse Connector does not support grouping");
    }
    return select(symbols, builder.toString());
}
```

openLookEng

ClickHouse

 openLookEng  
<https://openlookeng.io>

# ClickHouse connector 实现

- 函数映射
  - 直接映射
    - 常用函数：聚合统计函数、数学函数、字符串函、时间日期函数
    - 不满足业务需求

函数类型	函数语法	适配开发处理
聚合统计函数	CORR(\$1,\$2), STDDEV(\$1), stddev_pop(\$1), stddev_samp(\$1), skewness(\$1), kurtosis(\$1), VARIANCE(\$1), var_samp(\$1)	大多数基本一致，直接映射即可
数学函数	ACOS(\$1), ASIN(\$1), ATAN(\$1), ATAN2(\$1,\$2), CEIL(\$1), CEILING(\$1), COS(\$1), e(), EXP(\$1), FLOOR(\$1), LN(\$1), LOG10(\$1), LOG2(\$1), MOD(\$1,\$2), pi(), POW(\$1,\$2), POWER(\$1,\$2), RAND(), RANDOM(), ROUND(\$1), ROUND(\$1,\$2), SIGN(\$1), SIN(\$1), SQRT(\$1), TAN(\$1)	大多数基本一致，直接映射即可，部分函数需要对不同参数适配转换，分别有：1:n、n:1、n:m、1:m等类型。
字符串函数	CONCAT(\$1,\$2), LENGTH(\$1), LOWER(\$1), LTRIM(\$1), REPLACE(\$1,\$2), REPLACE(\$1,\$2,\$3), RTRIM(\$1), STRPOS(\$1,\$2), SUBSTR(\$1,\$2,\$3), POSITION(\$1,\$2), TRIM(\$1), UPPER(\$1)	大多数基本一致，直接映射即可，部分函数需要对不同参数适配转换，分别有：1:n、n:1、n:m、1:m等类型。
时间日期函数	YEAR(\$1), MONTH(\$1), QUARTER(\$1), WEEK(\$1), DAY(\$1), HOUR(\$1), MINUTE(\$1), SECOND(\$1), DAY_OF_WEEK(\$1), DAY_OF_MONTH(\$1), DAY_OF_YEAR(\$1)	直接映射

Code:

[https://github.com/Heatao/openLooKeng\\_ClickHouseConnector](https://github.com/Heatao/openLooKeng_ClickHouseConnector)

# | ClickHouse connector 实现

- 函数映射
  - v1.1 相似语义替换
  - 只能覆盖少部分函数
  - 语法不规范
  - 例子:

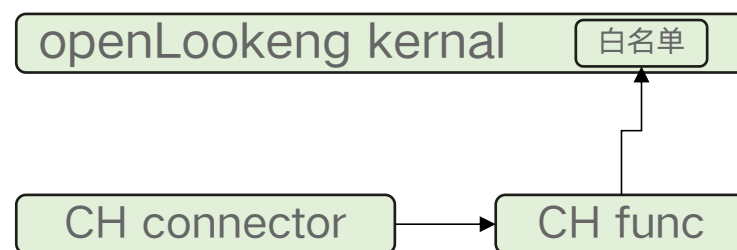
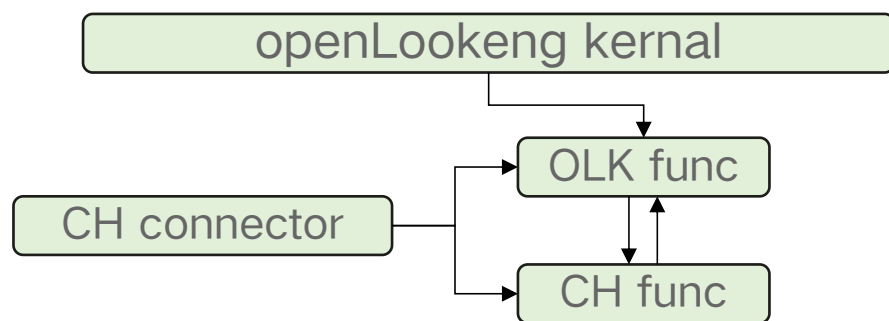
ClickHouse函数	openLooKeng函数	适配开发处理
toDateTime(\$1)	DATE_PARSE(\$1, \$2)	DATE_PARSE(\$1, '%Y-%m-%d %H:%i:%s')
toDate(\$1)	DATE_PARSE(\$1, \$2)	DARE_PARSE(\$1, '%Y-%m-%d')

Code:

[https://github.com/Heatao/openLooKeng\\_ClickHouseConnector](https://github.com/Heatao/openLooKeng_ClickHouseConnector)

# ClickHouse connector 实现

- 函数映射
  - v1.2 外部函数的注册下推
    - ClickHouseExternalFunctionHub#getExternalFunctions
    - ClickHouseExternalDateTimeFunctions#getFunctionsInfo
  - 扩展实现
    - ApplyRemoteFunctionPushDown#rewriteRemoteFunction
    - BaseJdbcRowExpressionConverter#visitCall



Ref: <https://openlookeng.io/zh-cn/docs/docs/develop/externalfunction-registration->

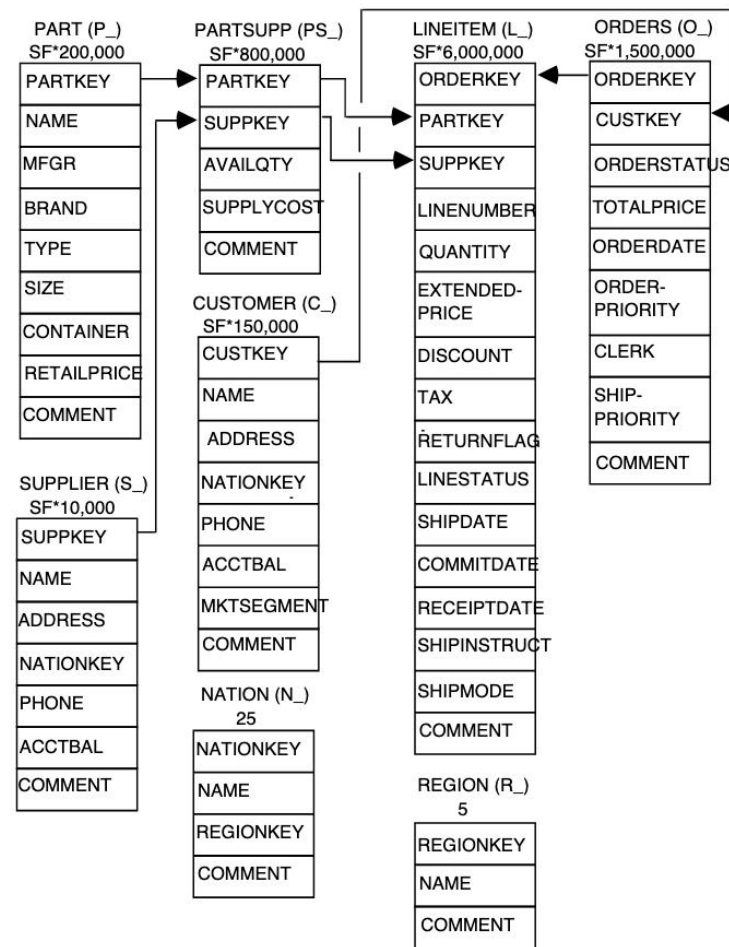
## | connector 目前的限制

- 类型支持暂不全面
  - 支持常见类型
  - Int128,Int256,UInt256,IPV4...
- 关于array的常用函数暂时不支持
- 单表查询性能具有一定的损耗

Ref: <https://openlookeng.io/zh-cn/docs/docs/develop/externalfunction-registration->

# CH connector性能测试

- **TPC** – Transaction process performance Council
  - 事务处理性能测试委员会
  - TPC不给出基准程序的代码，而只给出基准程序的标准规范
- **TPC-H**
  - 评价特定查询的决策支持能力，强调服务器在数据挖掘、分析处理方面的能力
  - 测试数据库系统复杂查询的响应时间，以每小时执行的查询数作为度量指标
  - 包含8张表，22个SQL查询
- **TPC-DS**
  - TPC-DS采用星型、雪花型等多维数据模式
  - 包含7张事实表，17张维度表平均每张表含有18列
  - 其工作负载包含99个SQL查询，覆盖SQL99和2003的核心部分以及OLAP
  - 这个测试集包含对大数据集的统计、报表生成、联机查询、数据挖掘等复杂应用



# CH connector性能测试

- SSB
  - Star Schema Benchmark
  - 基于TPC-H修改，并专门针对星型模型OLAP场景下的测试工具，共13条测试语句

clickhouse.tech/docs/en/getting-started/example-datasets/star-schema/

### Star Schema Benchmark

Getting Started / Example Datasets

#### Star Schema Benchmark

Compiling dbgen:

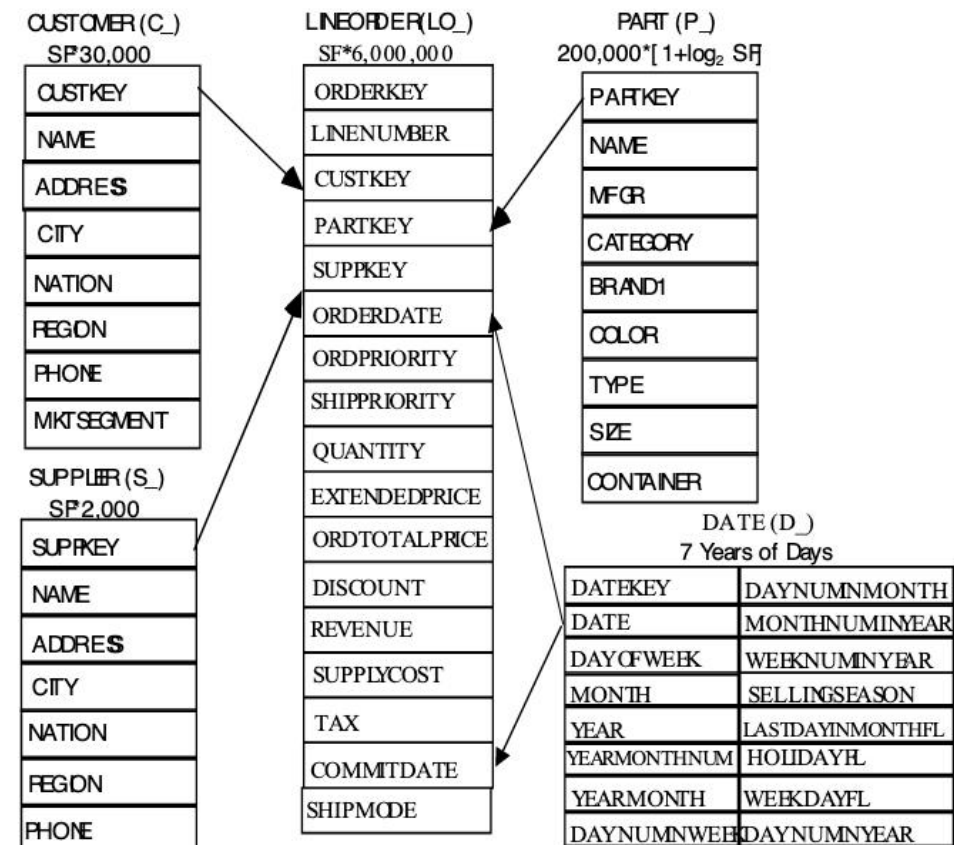
```
$ git clone git@github.com:vadimtk/ssb-dbgen.git
$ cd ssb-dbgen
$ make
```

Generating data:

**Attention**

With `-s 100` dbgen generates 600 million rows (67 GB), while while `-s 10`

```
$ ./dbgen -s 1000 -T c
$ ./dbgen -s 1000 -T l
$ ./dbgen -s 1000 -T p
$ ./dbgen -s 1000 -T s
```



Ref: <https://clickhouse.tech/docs/en/getting-started/example-datasets/star-schema>



# CH connector性能测试

- 使用ssb-dbgen工具构建了43亿条数据用于测试

table	size	bytes_on_disk	data_uncompressed_bytes	data_compressed_bytes	compress_rate	rows
customer	1.12 GiB	1.12 GiB	1.65 GiB	1.12 GiB	67.94630597	30000000
part	34.47 MiB	34.47 MiB	48.97 MiB	34.39 MiB	70.22835711	2000000
supplier	75.39 MiB	75.39 MiB	110.64 MiB	75.33 MiB	68.08099618	2000000
lineorder	123.00 GiB	123.00 GiB	176.19 GiB	122.76 GiB	69.67630518	4398761522
lineorder_flat	396.40 GiB	396.40 GiB	711.53 GiB	395.76 GiB	55.62024948	4398761522

# CH connector性能测试

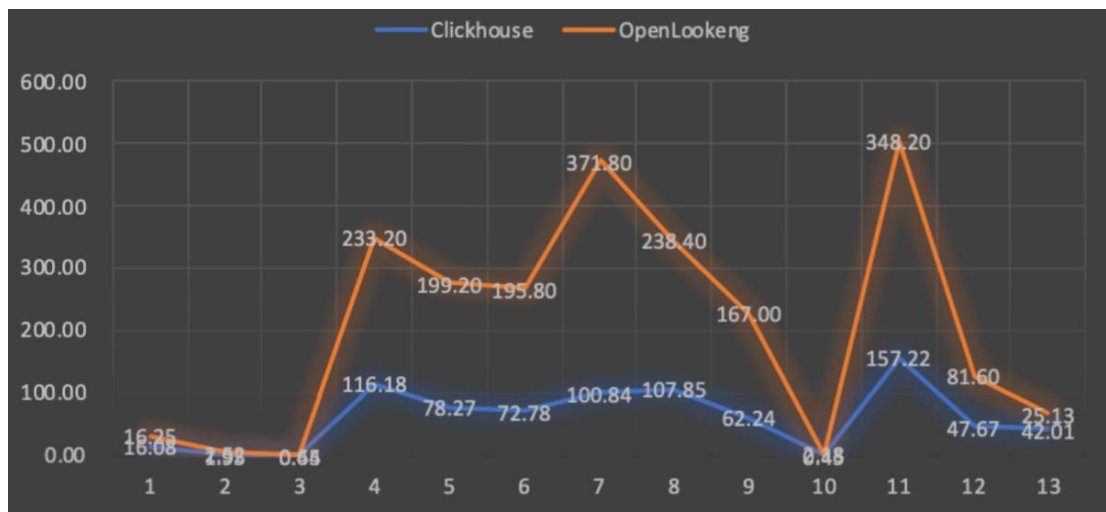
- SSB共13条测试用SQL语句
- 每条SQL语句执行间隔1分钟
- 执行多次取平均值

Q1.1	SELECT sum(LO_EXTENDEDPRICE * LO_DISCOUNT) AS revenue FROM lineorder_flat WHERE toYear(LO_ORDERDATE) = 1993 AND LO_DISCOUNT BETWEEN 1 AND 3 AND LO_QUANTITY < 25;	SELECT sum(LO_EXTENDEDPRICE * LO_DISCOUNT) AS revenue FROM lineorder_flat WHERE year(LO_ORDERDATE) = 1993 AND LO_DISCOUNT BETWEEN 1 AND 3 AND LO_QUANTITY < 25;
Q1.2	SELECT sum(LO_EXTENDEDPRICE * LO_DISCOUNT) AS revenue FROM lineorder_flat WHERE toYYYYMM(LO_ORDERDATE) = 199401 AND LO_DISCOUNT BETWEEN 4 AND 6 AND LO_QUANTITY BETWEEN 26 AND 35;	SELECT sum(LO_EXTENDEDPRICE * LO_DISCOUNT) AS revenue FROM lineorder_flat WHERE year(LO_ORDERDATE)*100 + month(LO_ORDERDATE) = 199401 AND LO_DISCOUNT BETWEEN 4 AND 6 AND LO_QUANTITY BETWEEN 26 AND 35;
Q1.3	SELECT sum(LO_EXTENDEDPRICE * LO_DISCOUNT) AS revenue FROM lineorder_flat WHERE toISOWeek(LO_ORDERDATE) = 6 AND toYear(LO_ORDERDATE) = 1994 AND LO_DISCOUNT BETWEEN 5 AND 7AND LO_QUANTITY BETWEEN 26 AND 35;	SELECT sum(LO_EXTENDEDPRICE * LO_DISCOUNT) AS revenue FROM lineorder_flat WHERE week(LO_ORDERDATE) = 6 AND year(LO_ORDERDATE) = 1994 AND LO_DISCOUNT BETWEEN 5 AND 7AND LO_QUANTITY BETWEEN 26 AND 35;

# CH connector性能测试

- 在1.0版本下，通过connector查询相比直接通过ClickHouse查询慢了134.08%

序号	Clickhouse	OpenLookeng	差异	百分比
Q1.1	16.08	16.25	-0.17	-1.05%
Q1.2	1.95	2.52	-0.57	-29.34%
Q1.3	0.44	0.65	-0.21	-48.44%
Q2.1	116.18	233.20	-117.02	-100.73%
Q2.2	78.27	199.20	-120.93	-154.50%
Q2.3	72.78	195.80	-123.02	-169.02%
Q3.1	100.84	371.80	-270.96	-268.71%
Q3.2	107.85	238.40	-130.55	-121.06%
Q3.3	62.24	167.00	-104.76	-168.30%
Q3.4	0.45	2.18	-1.73	-381.59%
Q4.1	157.22	348.20	-190.98	-121.47%
Q4.2	47.67	81.60	-33.93	-71.16%
Q4.3	42.01	25.13	16.88	40.18%
总计	803.98	1881.93	-1077.95	-134.08%



# CH connector性能测试

- 在1.2版本下，通过connector查询相比直接通过ClickHouse查询损耗降低，性能相比1.0版本有了显著提升
- 1.2版本给connector更多的权限，connector还可以进一步优化提升

序号	Clickhouse	OpenLookeng	差异	百分比
Q1.1	14.07	16.10	-2.03	-14.41%
Q1.2	1.55	2.41	-0.87	-55.99%
Q1.3	0.50	0.67	-0.17	-34.00%
Q2.1	119.83	169.40	-49.57	-41.36%
Q2.2	5.53	9.93	-4.40	-79.53%
Q2.3	3.97	8.98	-5.01	-126.15%
Q3.1	39.90	68.20	-28.30	-70.93%
Q3.2	39.80	38.72	1.08	2.71%
Q3.3	12.28	20.89	-8.61	-70.11%
Q3.4	0.23	1.82	-1.59	-694.76%
Q4.1	107.18	191.40	-84.22	-78.57%
Q4.2	17.81	22.63	-4.82	-27.07%
Q4.3	11.40	19.39	-8.00	-70.17%
总计	374.04	570.53	-196.49	-52.53%

## | 与Trino的ClickHouse connector的对比

- 对比Trino的ClickHouse connector，我们构建了一个仅有5000万条记录的表用于测试

序号	ClickHouse	openLookEng	Trino
Q1.1	0.775	2.80	22.23
Q1.2	1.22	1.84	18.93
Q1.3	0.094	0.373	10.19

## 未来计划

- Galaxy数据平台架构的重构
  - 接入更多数据源，非结构化的数据
  - 借助openLookEng的优势，提供更多方案选择
- CH connector 将在二周内提交到社区
- 欢迎交流：hezhengjie@iie.ac.cn



Thank you!

