

四川航空基于SelectDB的湖仓一体大数据 分析引擎建设

吴乐 信息技术部 大数据架构师

目录

01 公司介绍与业务背景

02 架构演进

03 SelectDB 湖仓一体应用实践

04 规划与展望

01

公司介绍与业务背景

四川航空股份有限公司

四川航空股份有限公司（以下简称：川航）以“美丽川航，美好出行”为使命，以“熊猫之路，连接世界”为愿景，以安全为品牌核心价值，自开航至今持续安全飞行36年，现运营全空客机队超过200架飞机，年运送旅客量超过3000万，航线网络覆盖亚洲、欧洲、北美洲、大洋洲和非洲，为全球旅客提供深具“中国元素，四川味道”的航空服务，品牌价值超过九百亿。



航司业务特点

业务系统多

- 业务复杂性和多样性：航空公司涉及的业务涵盖多个领域，包括航班调度、票务销售、旅客服务、机组管理、机务维修、财务结算等。
- 国际化与合规要求：航空公司运营跨国航班，需满足不同国家的法规和合规要求。

数据交互复杂

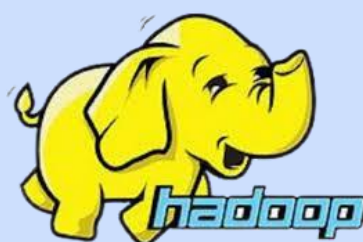
- 机场和航司需要共享大量的实时数据，确保航班的顺利运行。包括：航班动态信息、旅客信息、地面保障等信息。
- 航信数据交互，涉及到定票、航班信息、乘客离港等多个环节。

实时要求高

- 航班调度：确保航班按时起降，及时反映航班状态（如延误、取消），避免航班冲突和提高安全性。
- 乘客体验与服务：乘客信息（如登机、行李、座位安排等）需要实时更新，确保顺畅的登机流程

大数据平台建设历程

ORACLE



palo

SELECTDB

数据仓库与数据集市

1. 集中业务系统数据
2. 按需开发数据报表

大数据 Hadoop 平台

1. 存储力和算力大
2. 提升报表刷新时效

基于 Hadoop 和 Palo 的数据中台

1. 数据治理
2. 数据底座
3. 数据产品

湖仓一体分析引擎

2010

2014

2018

2022

2023

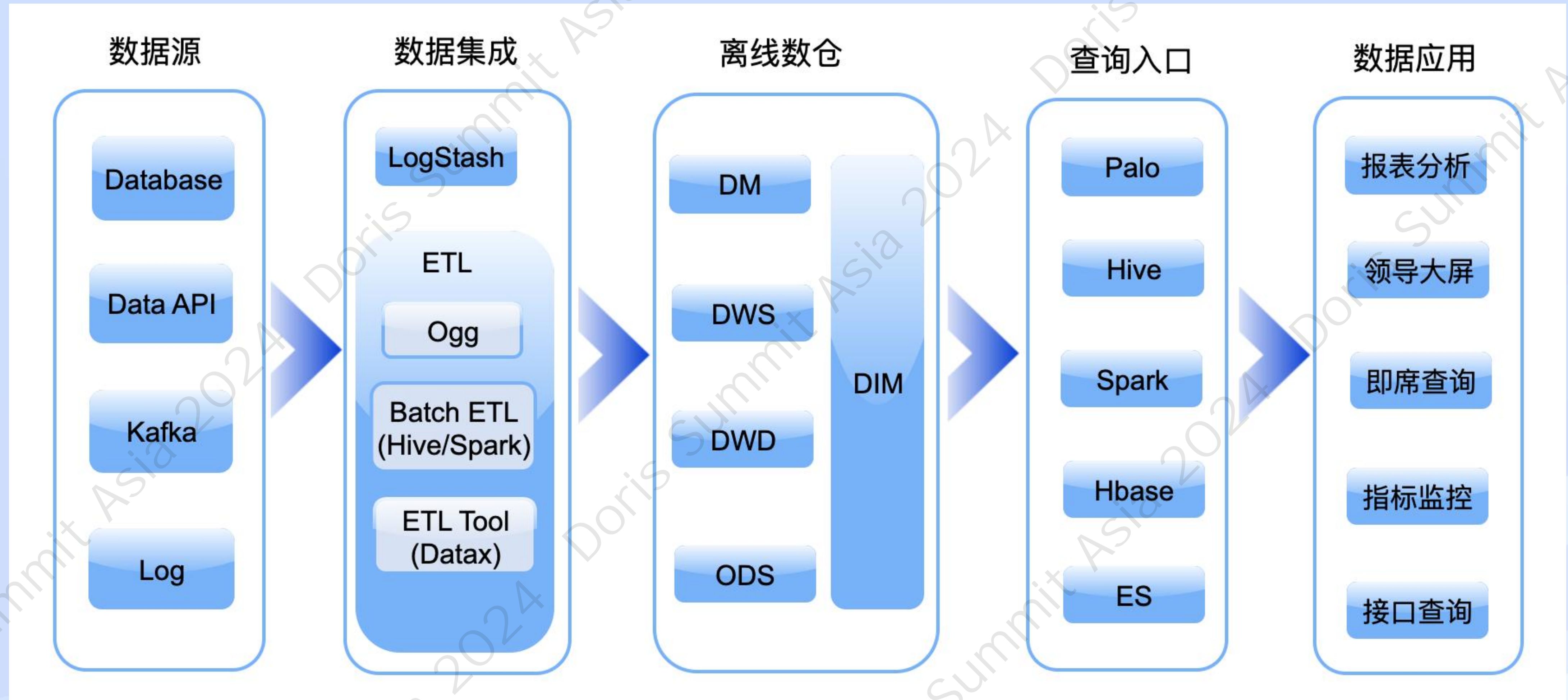
2024

统一 OLAP 技术栈、统一数据服务

02

架构演进

早期架构总览



痛点与需求

架构痛点

- 架构复杂，涉及组件较多，运维困难
- 支持高并发查询
- 数据时效性和查询效率低
- 大规模数据导入瓶颈
- 用户开发成本高

目标需求

- 架构简单，低运维成本
- 统一易用，降低用户学习成本
- 具备高效多数据源导入能力
- 秒级查询响应、快速扩展
- 实时高效，支持实时数据导入，湖仓加速实现高效查询

选型因素

支持高并发

SelectDB 在面对大数据量和高并发的情况下能够展现出出色的性能，可以替换 Hive 以及 Palo 数据源提供查询服务

Join 能力出色

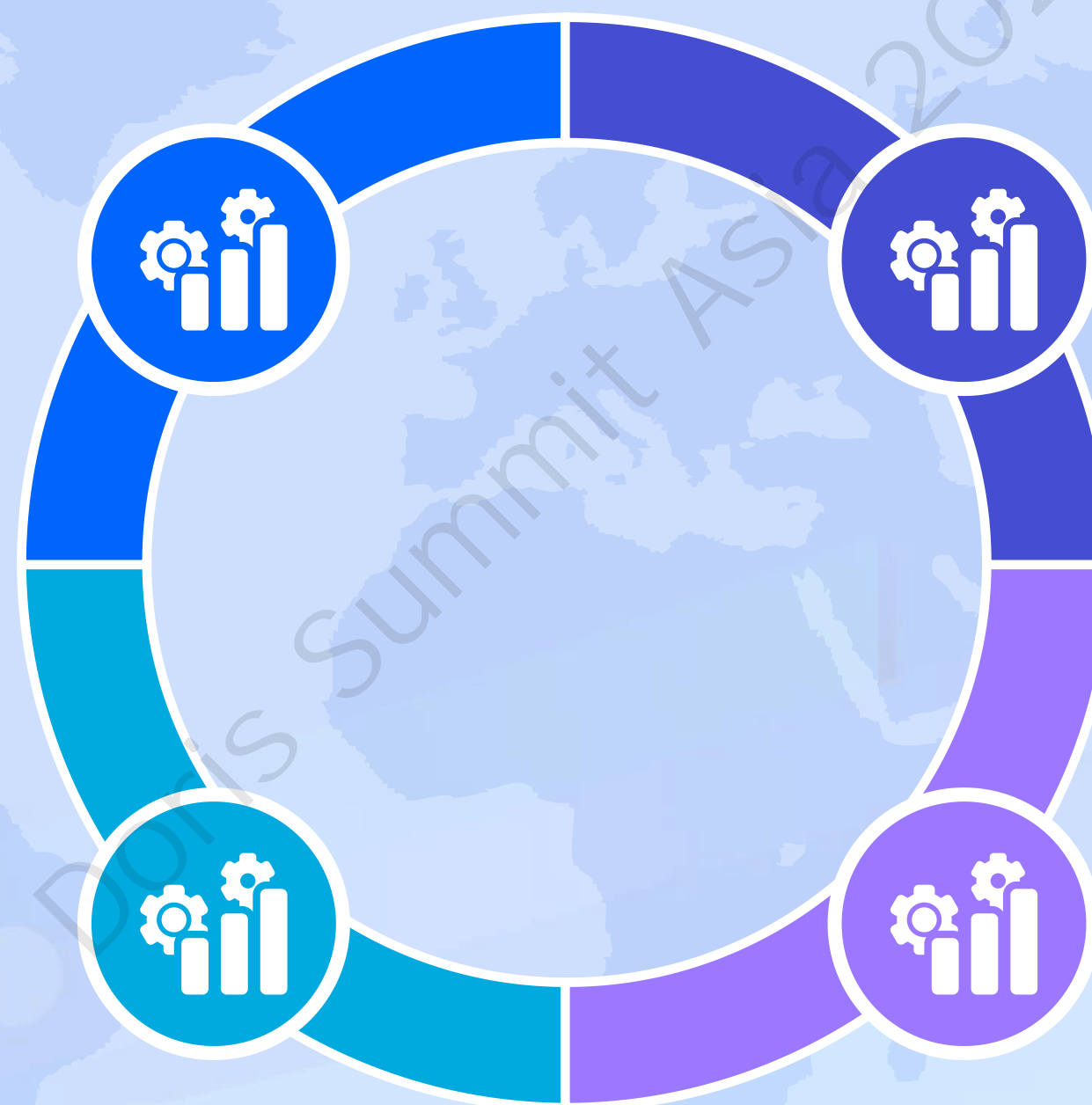
SelectDB 在处理大表的 Join 操作方面表现出色。支持多种分布式 Join 方式，包括 Broadcast Join、Shuffle Join、Hash Join 等方式

导入方式丰富

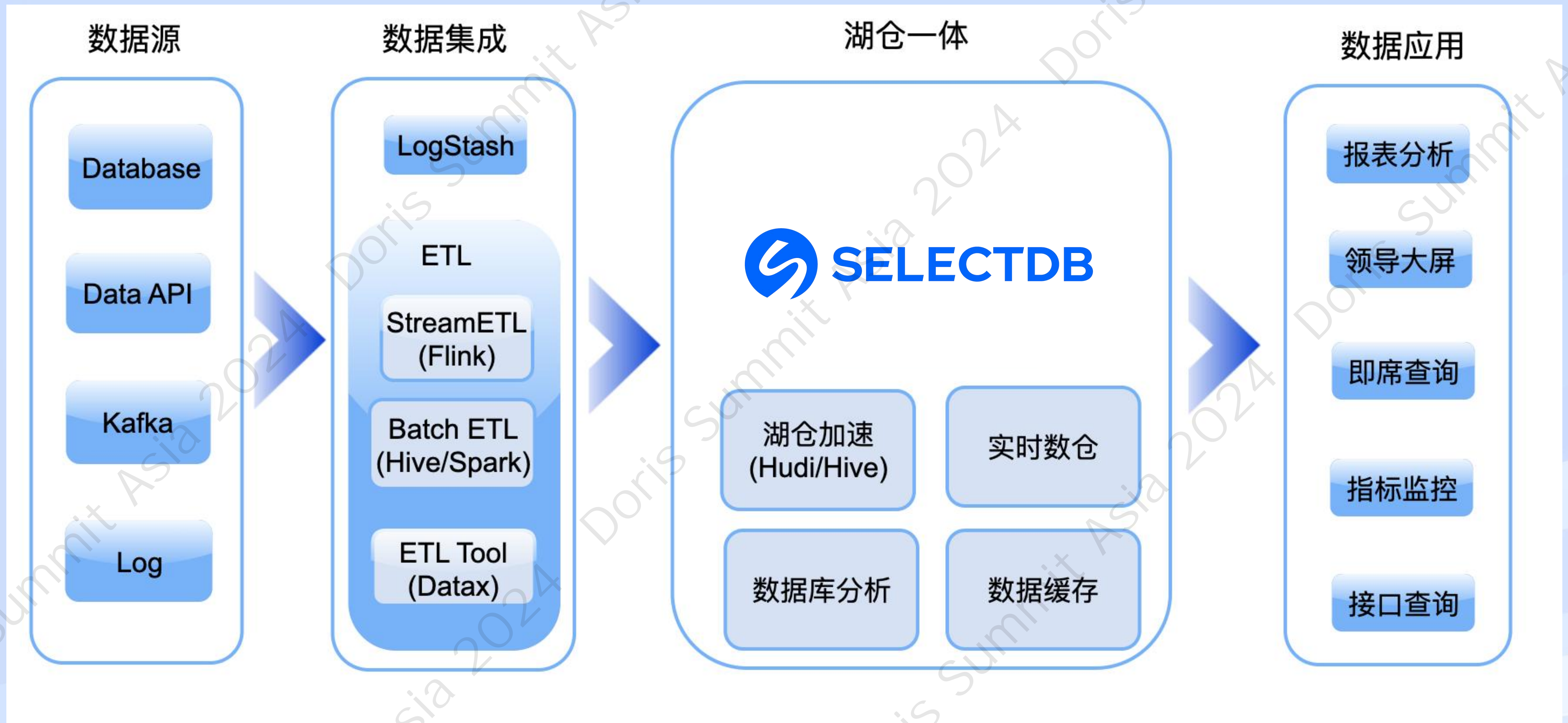
SelectDB 支持多种数据导入方式，包括 HDFS、Kafka、Spark、Flink 以及 RoutineLoad、BrokerLoad 和 StreamLoad，满足不同场景需求。

数据生态圈丰富

SelectDB 与 Spark、Flink 以及 DataX、SeaTunnel 等计算框架兼容，官方提供代码示例，易于使用。



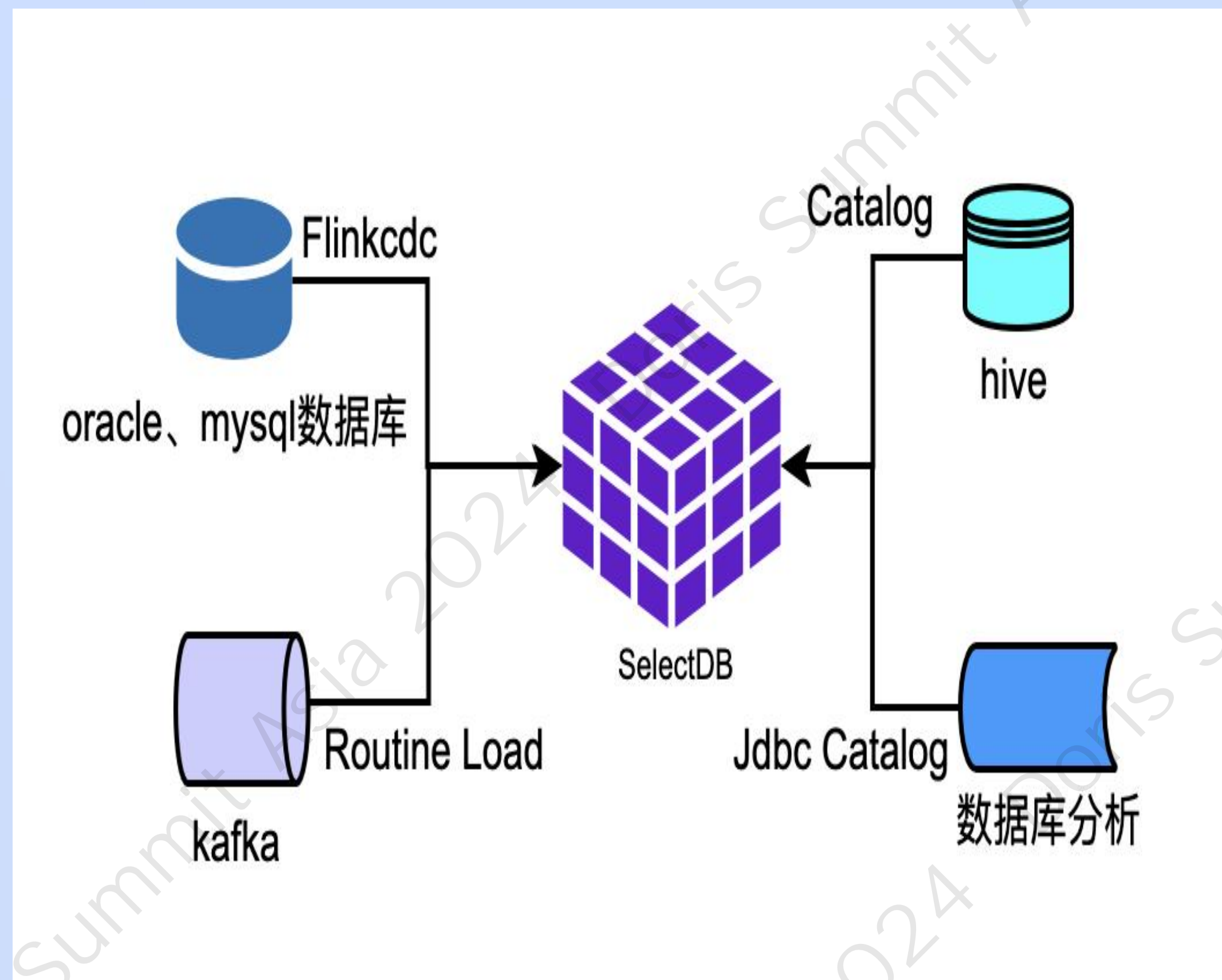
基于 SelectDB 构建的湖仓引擎



03

SelectDB 湖仓一体应用实践

多源数据快速接入



- **实时同步入湖**

对接核心数据库包括航班调度、旅客服务等，数据实时同步入湖。

- **对接消息队列**

机场的消息报文等数据，通过 Routine Load 对接 Kafka 数据，消费数据入湖。

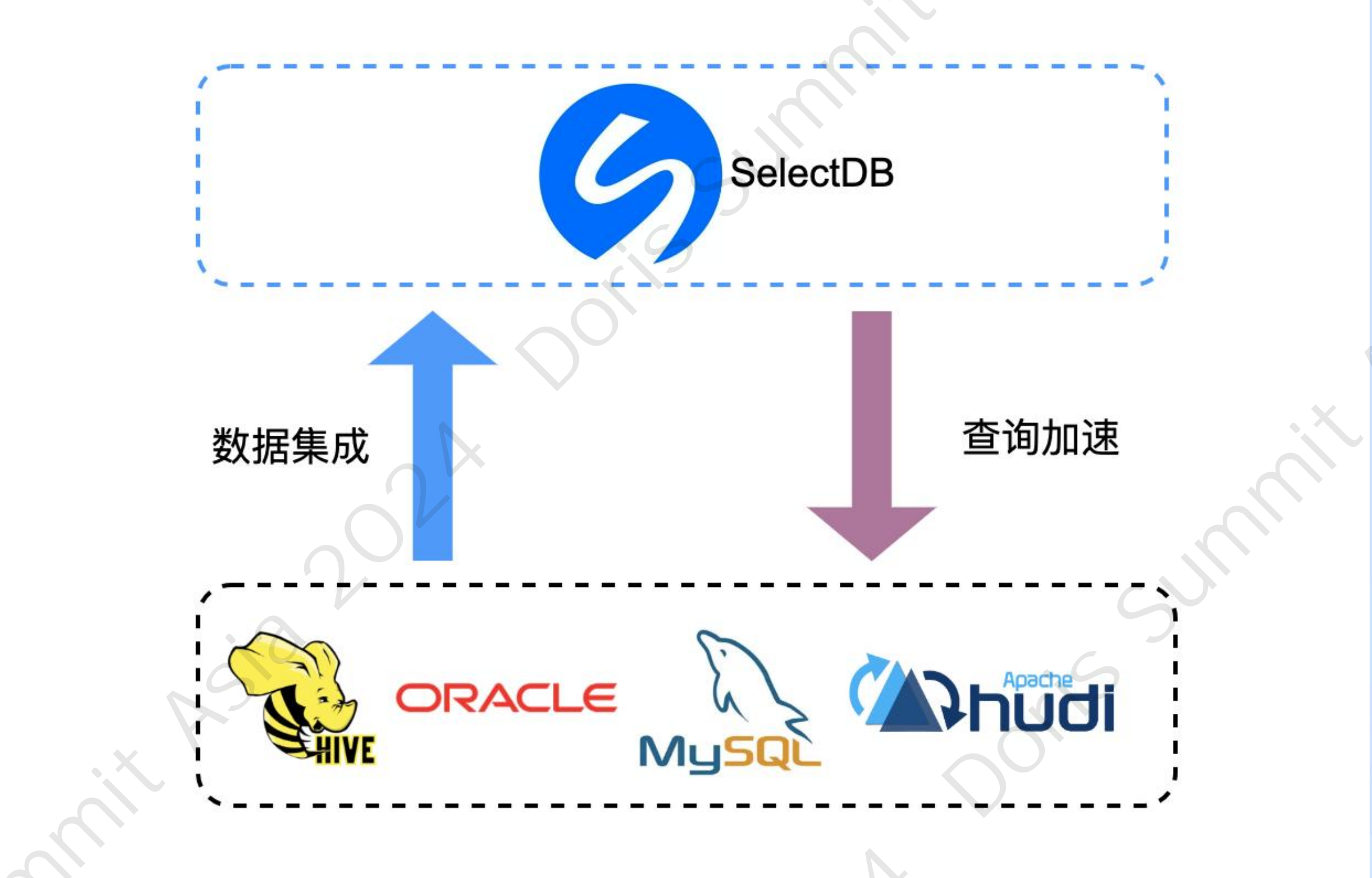
- **Hive 加速查询**

Hive 主要存储 acars 日志数据，通过 Hive Catalog 对接，方便和 SelectDB 内表关联，加速查询。

- **数据库分析**

针对一些不常用的数据库，通过外表查询，快速获取数据。

基于 SelectDB 的联邦数据湖分析场景



```
CREATA CATALOG HIVE
PROPERTIES(
  'type'='hms','hive.metastore.uris'='thrift://172.0.0.1:9083',
);

SELECT * FROM HIVE.DB.TABLE a JOIN INTERNAL.DB.TABLE b
ON a.id =b.id;

INSERT INTO INTERNAL.DB.TABLE SELECT * FROM HIVE.DB.TABLE
WHERE DATE="2024-11-21";

INSERT INTO JDBC.DB.TABLE1 SELECT *FROM HIVE.DB.TABLE
WHERE DATE="2024-11-21";
```

SQL 查询分析 数据导入 数据写回

高性能更新能力

```
CREATE TABLE ODS_SCALFOC.`ODS_FLIGHT_BASIC_INFO` (  
  `FLIGHT_ID` decimal(9,0) NULL COMMENT '航班主键',  
  `FLIGHT_DATE` date NULL COMMENT '航班日期',  
  `FLIGHT_TYPE` varchar(23) NULL COMMENT '航班类型',  
  `FLIGHT_NO` varchar(10) NULL COMMENT '航班号',  
  `AC_TYPE` varchar(50) NULL COMMENT '机型',  
  `LAYOUT` varchar(50) NULL COMMENT '布局',  
  `AC_REG` varchar(50) NULL COMMENT '机号',  
  `DEPARTURE_AIRPORT` varchar(32) NULL COMMENT '起飞机场',  
  `ARRIVAL_AIRPORT` varchar(32) NULL COMMENT '目的机场',  
  `HTD` datetime NULL COMMENT 'HTD',  
  `STD` datetime NULL COMMENT '预飞',  
  `ETD` datetime NULL COMMENT '计飞',  
  `OUT` datetime NULL COMMENT '滑出',  
  .....  
) ENGINE=OLAP  
UNIQUE KEY(`FLIGHT_ID`, `FLIGHT_DATE`)  
COMMENT 'OLAP'  
DISTRIBUTED BY HASH(`FLIGHT_ID`) BUCKETS 10  
PROPERTIES (  
  "replication_allocation" = "tag.location.default: 3",  
  "enable_unique_key_merge_on_write" = "true"  
);
```

MERGE ON WRITE 更新能力

适用场景：

适用于小批量实时高频导入，基于主键做高频数据更新
目前数仓 ODS 层数据是 CDC 实时同步导入，全部采用 MOR 表，同步数据为根据主键进行 UPSERT。

测试数据约 500G，常规查询以及复杂关联等，使用 MOW 模型，性能提升近 4 倍

50 并发	平均耗时
MOR	4.6
MOW	1.2
提升	3.9 倍

部分列更新场景

业务场景	事件名
离港航班变更	离港航初始化
	离港航节机型变更
	离港航节起飞时间变更
	离港航班状态变更
	离港航班登机口变更

原数据：

FlightNumber	FlightDate	DepAirport	ArrAirport	FlightStatus	Deptime
3U3828	2024-11-03	TFU	KMG	CI	08:30
3U3281	2024-11-04	CTU	ALA	CI	20:30

新增：

FlightNumber	FlightDate	DepAirport	ArrAirport	FlightStatus
3U3281	2024-11-04	CTU	ALA	CC

航班状态变更报文样例片段

```
{
  "Hdr": {
    "Ver": "v1.0",
    "Event": "StatusChange",
    "Subevent": "CC",
    "Newval": "CC",
    "Oldval": "CI",
    "Uptm": "20241104131901310",
    "Stamp": "0b672c44-7c58-46d8-a565-e8516eec4568"
  },
  "Dat": {
    "Flight": {
      "AirlineCode": "3U",
      "FlightNumber": "9821",
      "International": "I",
      "Route": "CTU-ALA-BUD",
      "FlightDate": "2024-11-04",
      "FlightStatus": "CC",
      "Suffix": "",
      "DepAirport": "CTU",
      "ArrAirport": "ALA"
    }
  }
}
```

Flink 设置部分列更新

'sink.properties.partial_columns' = 'true'
在 sink.properties.column 中指定要导入的列（必须包含所有 key 列，不然无法更新）

合并后：

FlightNumber	FlightDate	DepAirport	ArrAirport	FlightStatus	Deptime
3U3828	2024-11-03	TFU	KMG	CI	08:30
3U3281	2024-11-04	CTU	ALA	CC	20:30

数据建模场景

AGGREGATE KEY 报表统计、指标计算

```
CREATE TABLE selectdb_agg_tab(  
  flight_id varchar(30),  
  date varchar(30),  
  ac_cnt BIGINT SUM DEFAULT '0'  
) AGGREGATE KEY(flight_id, date)  
DISTRIBUTED BY HASH(flight_id) BUCKETS  
  10
```

原表:

flight_id	date	ac_cnt
1001	11-15	110
1002	11-15	80

新增:

flight_id	date	ac_cnt
1002	11-15	60
1003	11-15	130

结果:

flight_id	date	ac_cnt
1001	11-15	110
1002	11-15	140
1003	11-15	130

UNIQUE KEY 航班状态、客票状态

```
CREATE TABLE selectdb_uni_tab(  
  flight_id varchar(30),  
  date varchar(30),  
  status varchar(30),  
) UNIQUE KEY(flight_id)  
DISTRIBUTED BY HASH(flight_id) BUCKETS  
  10
```

原表:

flight_id	date	status
1001	11-15	起飞
1002	11-15	计划

新增:

flight_id	date	status
1002	11-15	起飞

结果:

flight_id	date	status
1001	11-15	计划
1002	11-15	起飞

DUPLICATE KEY 明细数据、变更记录

```
CREATE TABLE selectdb_dup_tab(  
  tkt_id varchar(30),  
  date varchar(30),  
  status varchar(30),  
) DUPLICATE KEY(tkt_id)  
DISTRIBUTED BY HASH(tkt_id) BUCKETS 10
```

原表:

tkt_id	date	status
101	11-15	待使用
102	11-15	待值机

新增:

flight_id	date	status
102	11-15	已值机

结果:

flight_id	date	ststus
101	11-15	待使用
102	11-15	待值机
102	11-15	已值机

SelectDB 极速分析场景



倒排索引

倒排索引实现多维度快速检索分析，
加速字符串类型数据的全文检索、支持自定义分词

```
CREATE INDEX IDX_CONTENT ON QUESTION(`REPLY_CONTENT`) USING  
INVERTED PROPERTIES("parser" = "chinese")
```

-- 查询评价内容里包含好评

```
select * from QUESTION where REPLY_CONTENT match '好评'
```

-- 分词分析

```
SELECT TOKENIZE(REPLY_CONTENT,"parser"="chinese","parser_mode"=  
"fine_grained") FROM QUESTION
```

100并发	V1.2.1	V2.0.15（倒排索引）
SQL1平均耗时	1890	463
SQL2平均耗时	793	112
SQL3平均耗时	523	247
总计耗时(ms)	3206	822

查询性能提升约 4 倍

SelectDB 湖仓应用收益

5-13 倍

ETL

3-6 倍

数据导入

10-18 倍

查询分析

5 秒内

实时

1000 W+次

单日查询量

<1 秒 (96%)

查询延时

04

规划与展望

未来规划

架构优化

- 存算分离架构，引入更为廉价的存储介质以降低成本，数据湖场景下更灵活地弹性部署。
- CCR 主备集群同步优化，探索主备集群实现自动负载，数据自动恢复。

统一日志分析

- 利用 SelectDB 倒排索引替换目前 ES 做日志分析，降低成本。

自动化运维

- 引用 Manager 接管生产集群运维。
- 数据血缘分析
- 表热度分析、慢查询自动监控

Thanks for Watching!