

# Apache Doris

## 在有色工业互联网领域的应用

曹志富

中铝视拓 大数据架构师

# 目录

1. 背景

2. 演进

3. 实践

4. 展望

# 1 背景



# 中铝视拓



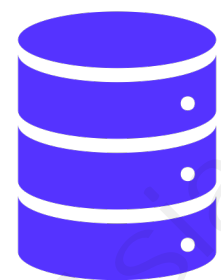
## 简介

中铝旗下混合所有制企业  
智能化专业队伍  
聚焦为有色金属工业提供智能制造  
整体解决方案

## 产品

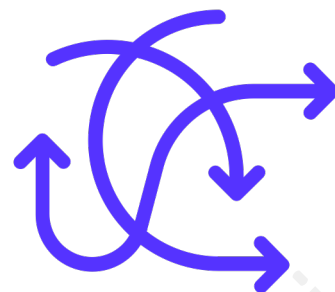
有色工业互联网通用PaaS平台  
数据平台  
数字工序  
智能管理  
智能决策

# 业务背景



## 类型多

研发数据  
生产数据  
管理数据  
安环数据



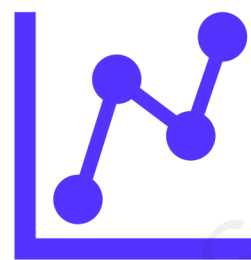
## 无标准

术语不一，计量不一  
设备协议不同，数值差异大



## 共享难

分布在多个系统  
系统之间无联通

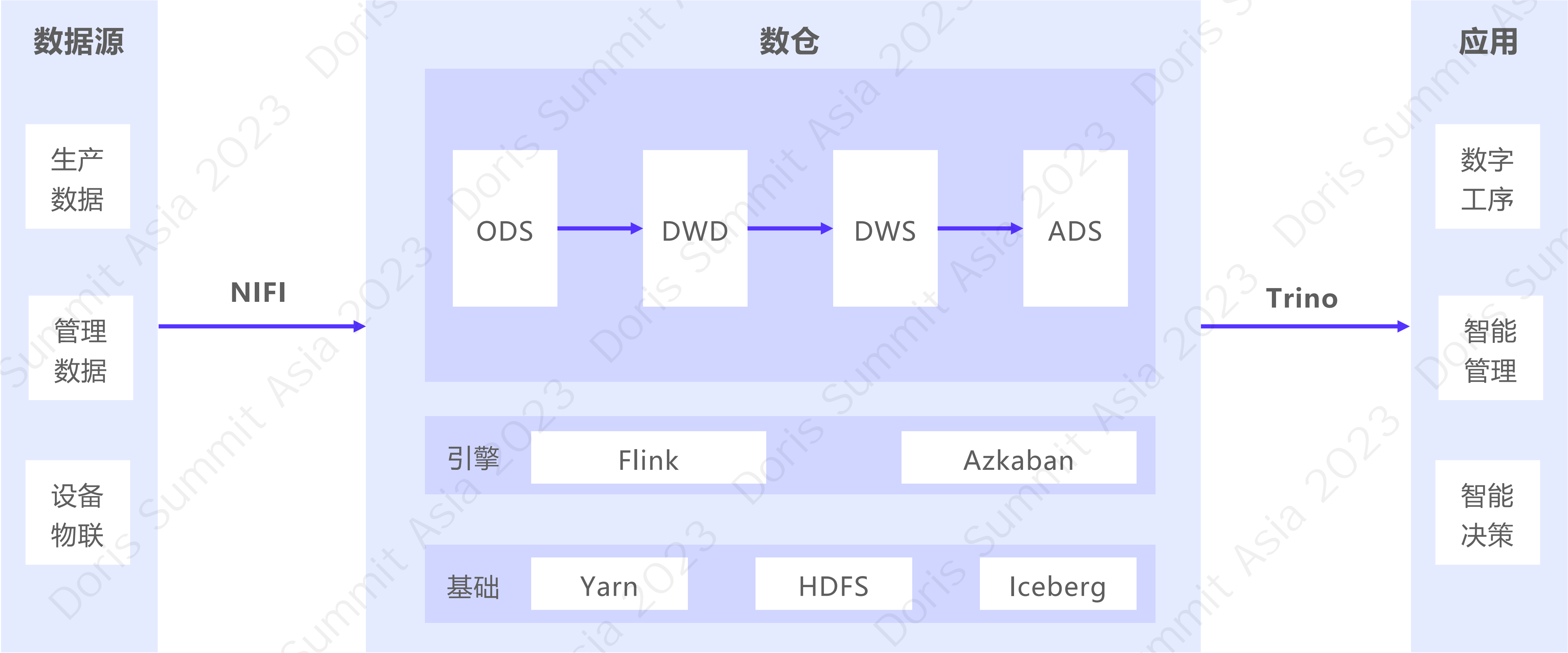


## 分析难

模型开发不足，模型应用、知识分享不足  
缺缺少深度挖掘分析，少深度挖掘分析

## 2 演进

# Flink + Iceberg



成效

标准	指标	应用
<ul style="list-style-type: none"><li>完成氧化铝、电解铝等多个板块标准制定</li><li>12 个维度类</li><li>1500+ 个铝工业指标的标准</li></ul>	<ul style="list-style-type: none"><li>电解铝 1500+ 个</li><li>氧化铝 600+ 个</li></ul>	<ul style="list-style-type: none"><li>总部平台：接入 30+ 企业数据</li><li>成功应用至电解铝、氧化铝标杆企业</li></ul>



困扰



资源高

工厂侧 10+ 节点



性能差

单任务分钟级  
并发任务 10+  
时序数据支持不够  
即席查询慢



运维难

小文件  
专业人员支持  
故障频发

# 新的选择：Doris



资源高



性能差



运维难



组件丰富

多源数据目录  
UDF



性能强劲

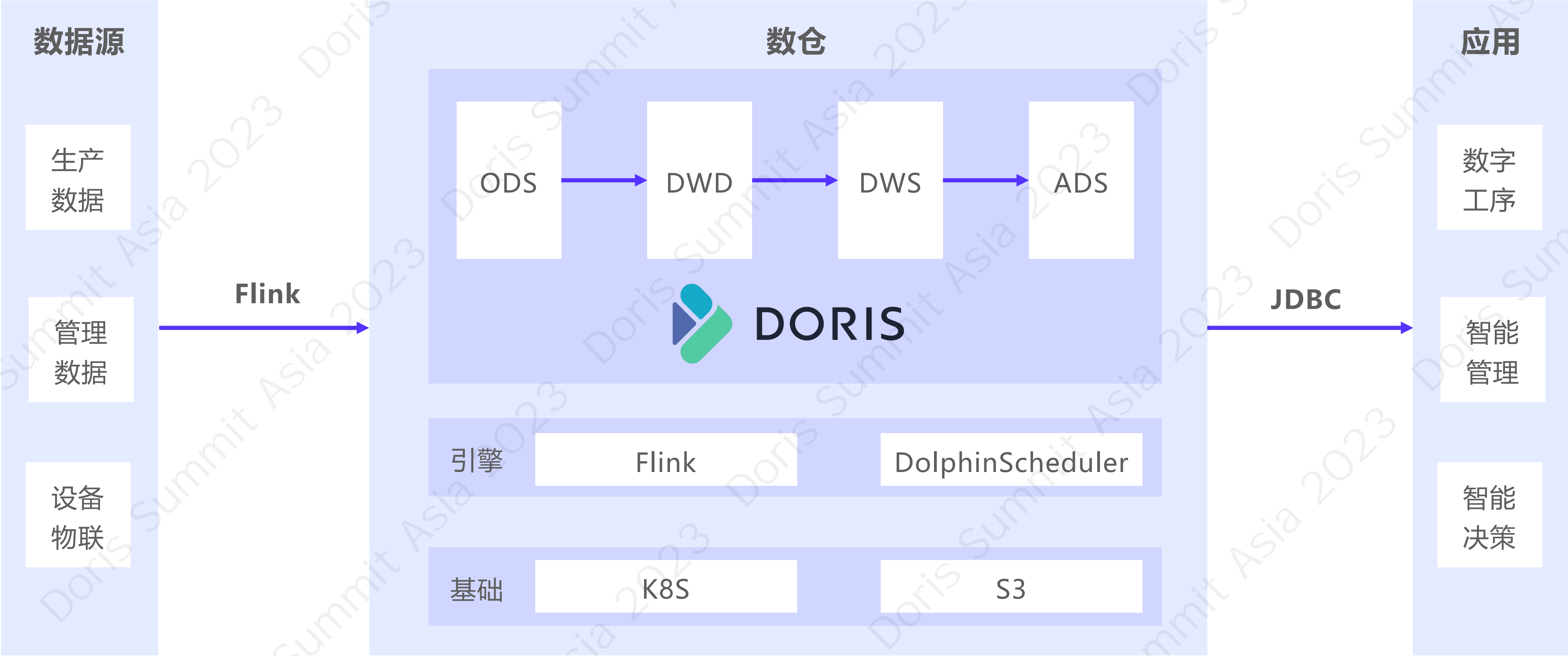
向量化执行引擎  
ZSTD



运维简单

FE + BE，无外部系统依赖  
无小文件困扰

# 新的选择：Doris 为基座



收益

资源需求下降

60%

平均作业耗时

1s

查询性能提升

10倍

月平均故障次数

0次



# 3 实践

设备数据分析

压滤机

批量构建表

批量构建作业

批量更新表

批量更新作业

批量配置优先级

批量删除

类型

名称

复合指标 压滤机\_班\_上一班循环次数

复合指标 压滤机\_班\_月累计循环次数

复合指标 企业\_班\_压滤机循环次数

复合指标 压滤机\_班\_循环次数

复合指标 赤泥压滤机\_日\_上一日夜班...

双生指标 赤泥压滤机\_日\_0点前夜班...

复合指标 赤泥压滤机\_日\_上一日4点...

双生指标 赤泥压滤机\_日\_4点前夜班...

双生指标 赤泥压滤机\_日\_夜班运转台时

复合指标 赤泥压滤机\_日\_上一日4点...

复合指标 企业\_班\_上一班压滤机循环...

双生指标 压滤机\_班\_丁班运转台时

双生指标 压滤机\_班\_丙班运转台时

双生指标 压滤机\_班\_乙班运转台时

双生指标 压滤机\_班\_甲班运转台时

复合指标 企业\_年\_压滤机运行效率

复合指标 企业\_月\_上一月压滤机运行...

复合指标 企业\_月\_压滤机运行效率

复合指标 赤泥压滤机\_日\_0点前循环...

复合指标 赤泥压滤机\_日\_4点前循环...

复合指标 赤泥压滤机\_日\_4点前夜班...

双生指标 赤泥压滤机\_日\_0点前循环...

复合指标 企业\_月\_上一月压滤机运转...

复合指标 压滤机\_年\_运转率

复合指标 压滤机\_年\_运转台时

复合指标 企业\_班\_丁班压滤机循环次数

复合指标 企业\_班\_丙班压滤机循环次数

复合指标 企业\_班\_甲班压滤机循环次数

复合指标 赤泥压滤机\_日\_上一日夜班...

复合指标 赤泥压滤机\_日\_上一日4点前夜...

双生指标 赤泥压滤机\_日\_0点前夜班...

初始化

已构建

数据预览 元数据

基本信息

计算逻辑

名称	压滤机_班_丙班运转台时	类型	一级衍生指标	编码	YALVJ_B_BBYZHTSH
标准编码	607158	标签	设备指标,赤泥压滤工序	版本	1.0.2
业务活动组合	赤泥压滤机1小时运行状态_轮班表	指标	运转台时	统计对象	压滤机
关联对象	-	统计周期	班	单位	h
优先级	以修正后数据为准	创建人	liuyanjin	创建时间	2023-08-10 16:54:36
修改人	liuyanjin	修改时间	2023-08-13 15:54:39	数据生产等级	-

公式1

计算逻辑

业务限定

对象属性

CASE WHEN SUM( 运转台时 ) >=28680 THEN 8 ELSE ( SUM( 运转台时 ) / 3600 ) END

班别 = '丙班'

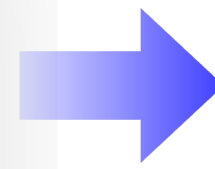
对象实例



```
insert into
GUANGXIHUASHENG.G.CHINIYLJ1XSHYXZHT_LB8
WITH LABEL 'label3c4c62a04491403690cdef9ac99819e5' with base as (
select
S_CHINIYLJ1XSHYXZHT.D_QIYE_BIANHAO as D_QIYE_BIANHAO,
'D_YALVJ' as ENTITY_TYPE,
S_CHINIYLJ1XSHYXZHT.D_YALVJ_BIANHAO as ENTITY_CODE,
'work_dim' as PERIOD_TYPE,
'work_dim' as PERIOD_CODE,
union_udf.custom_work(S_CHINIYLJ1XSHYXZHT.YEWUSHJ, 3) as PERIOD_TIME,
'班别_丙班' as COND_CODE,
S_CHINIYLJ1XSHYXZHT.D_QIYE_MINGCHENG as D_QIYE_MINGCHENG,
S_CHINIYLJ1XSHYXZHT.D_YALVJ_MINGCHENG as ENTITY_CNAME,
'班' as PERIOD_NAME,
'压滤机' as ENTITY_NAME
from
GUANGXIHUASHENG.S_CHINIYLJ1XSHYXZHT
inner join GUANGXIHUASHENG.S_PAIBANB
on DATE_FORMAT(S_PAIBANB.YEWUSHJ, '%Y%m%d:%H') = DATE_FORMAT(S_CHINIYLJ1XSHYXZHT.YEWUSHJ,
'%Y%m%d:%H')
where
S_CHINIYLJ1XSHYXZHT.YEWUSHJ >= union_udf.date_sub(now(3), 3)
group by
S_CHINIYLJ1XSHYXZHT.D_YALVJ_BIANHAO,
union_udf.custom_work(S_CHINIYLJ1XSHYXZHT.YEWUSHJ, 3),
S_CHINIYLJ1XSHYXZHT.D_QIYE_BIANHAO,
S_CHINIYLJ1XSHYXZHT.D_QIYE_MINGCHENG,
S_CHINIYLJ1XSHYXZHT.D_YALVJ_MINGCHENG
),
T0 as (
select
S_CHINIYLJ1XSHYXZHT.D_QIYE_BIANHAO as D_QIYE_BIANHAO,
'D_YALVJ' as ENTITY_TYPE,
S_CHINIYLJ1XSHYXZHT.D_YALVJ_BIANHAO as ENTITY_CODE,
'work_dim' as PERIOD_TYPE,
'work_dim' as PERIOD_CODE,
union_udf.custom_work(S_CHINIYLJ1XSHYXZHT.YEWUSHJ, 3) as PERIOD_TIME,
'班别_丙班' as COND_CODE,
S_CHINIYLJ1XSHYXZHT.D_QIYE_MINGCHENG as D_QIYE_MINGCHENG,
S_CHINIYLJ1XSHYXZHT.D_YALVJ_MINGCHENG as ENTITY_CNAME,
'班' as PERIOD_NAME,
'压滤机' as ENTITY_NAME,
case
when SUM(S_CHINIYLJ1XSHYXZHT.YUNZHUAHTSH) >= 28680 then 8
else (SUM(S_CHINIYLJ1XSHYXZHT.YUNZHUAHTSH) / 3600)
end as BANYUNZHTSH
from
GUANGXIHUASHENG.S_CHINIYLJ1XSHYXZHT
inner join GUANGXIHUASHENG.S_PAIBANB
on DATE_FORMAT(S_PAIBANB.YEWUSHJ, '%Y%m%d:%H') = DATE_FORMAT(S_CHINIYLJ1XSHYXZHT.YEWUSHJ,
'%Y%m%d:%H')
where
S_PAIBANB.BANBIE = '丙班'
and S_CHINIYLJ1XSHYXZHT.YEWUSHJ >= union_udf.date_sub(now(3), 3)
group by
S_CHINIYLJ1XSHYXZHT.D_YALVJ_BIANHAO,
union_udf.custom_work(S_CHINIYLJ1XSHYXZHT.YEWUSHJ, 3),
S_CHINIYLJ1XSHYXZHT.D_QIYE_BIANHAO,
S_CHINIYLJ1XSHYXZHT.D_QIYE_MINGCHENG,
S_CHINIYLJ1XSHYXZHT.D_YALVJ_MINGCHENG
)
select
base.D_QIYE_BIANHAO,
base.ENTITY_TYPE,
base.ENTITY_CODE,
base.PERIOD_TYPE,
base.PERIOD_CODE,
base.PERIOD_TIME,
base.COND_CODE,
base.D_QIYE_MINGCHENG,
base.ENTITY_CNAME,
base.PERIOD_NAME,
base.ENTITY_NAME,
now(3),
NULL,
T0.BANYUNZHTSH,
NULL,
NULL
from
base
left join T0 on base.PERIOD_TIME = T0.PERIOD_TIME
and base.ENTITY_CODE = T0.ENTITY_CODE
and base.COND_CODE = T0.COND_CODE;
```

## 多源数据目录

```
CREATE CATALOG oracle_ods_8_catalog PROPERTIES (  
  "type"="jdbc",  
  "user"="user",  
  "password"="123456",  
  "jdbc_url"="jdbc:oracle:thin:@//127.0.0.1:1521/orcl1",  
  "driver_url"="ojdbc8.jar",  
  "driver_class"="oracle.jdbc.driver.OracleDriver"  
);
```

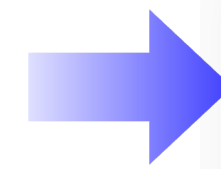


```
INSERT INTO datacatalog192.testdb.orcl_test_pg (  
  id,  
  pyuansdzhjrl,  
  mnysdzhjrl,  
  create_da,  
  comm,  
  create_date,  
  ddate,  
  ddatetime)  
VALUES ('55', '1', '2', '2023-07-06 00:00:00', 'oracle'  
  , '2023-08-06 00:00:00', '2023-08-06 00:00:00', '2023-08-06 00:00:00');
```



## UDF

```
StringBuilder sb = new StringBuilder();
int year = dateTime.getYear();
sb.append(year);
int month = dateTime.getMonth().getValue();
if (month < 10) {
    sb.append("0").append(month);
} else {
    sb.append(month);
}
int dayOfMonth = dateTime.getDayOfMonth();
if (dayOfMonth < 10) {
    sb.append("0").append(dayOfMonth);
} else {
    sb.append(dayOfMonth);
}
sb.append(":");
int hour = dateTime.getHour();
int hoursPerWork = 24 / numOfWork;
for (int i = 0; i < numOfWork; i++) {
    if (hour >= ((i) * hoursPerWork) && hour < ((i + 1) * hoursPerWork)) {
        sb.append(i);
    }
}
return sb.toString();
```



```
delete from
    test.S_CUYEC1XSHYXZHT
where BANCHI >
union_udf.custom_work(now(3))
```

```
CREATE ALIAS FUNCTION union_udf.custom_work(DATETIMEV2(3),int)
with PARAMETER (datetime1,int1) as
    concat(
        DATE_FORMAT(date_trunc(datetime1,'day'), '%Y%m%d'), ': ',
        multiply(floor(divide(HOUR(datetime1), divide(24,int1))),1));
```



## 设备表自动分区

```
CREATE TABLE `WH_GUANGXIHUASHENGSHEDIKONGJIAN` (  
  `BIANHAO` varchar(255) NULL ,  
  `DONGTAISHX` varchar(255) NULL ,  
  `YEWUSHJ` datetimev2(3) NULL ,  
  `SHUJUCHLSHJ` datetimev2(3) NULL ,  
  `ZHI` varchar(255) NULL ,  
  `TYPE` SMALLINT NULL  
) ENGINE = OLAP  
UNIQUE KEY(`BIANHAO`, `DONGTAISHX`, `YEWUSHJ` )  
PARTITION BY RANGE(`YEWUSHJ`) ()  
DISTRIBUTED BY HASH(`BIANHAO`) BUCKETS 3 PROPERTIES (  
  "replication_allocation" = "tag.location.default: 3",  
  "dynamic_partition.enable" = "true",  
  "dynamic_partition.time_unit" = "DAY",  
  "dynamic_partition.start" = "-7",  
  "dynamic_partition.end" = "3",  
  "dynamic_partition.prefix" = "ts",  
  "dynamic_partition.buckets" = "32",  
  "dynamic_partition.create_history_partition" =  
  "true",  
  "dynamic_partition.history_partition_num" = "10"  
);
```

## S3 Load 备份迁移

```
CREATE REPOSITORY s3_repo WITH S3
ON LOCATION "s3://ws.doris.backup/2023-05" PROPERTIES
(
  "AWS_ENDPOINT" = "http://127.0.0.1:15232",
  "AWS_ACCESS_KEY" = "ak",
  "AWS_SECRET_KEY" = "sk",
  "AWS_REGION" = "REGION",
  "use_path_style" = "true")
```

```
BACKUP SNAPSHOT ws.snapshot_label1
TO s3_repo
PROPERTIES ("type" = "full");
```

```
RESTORE SNAPSHOT ws.`snapshot_label1` FROM `s3_repo` PROPERTIES
("backup_timestamp" = "2023-07-20-14-34-53");
```



# 基于 Storage policy 的生命周期管理

```
CREATE RESOURCE "remote_s3" PROPERTIES
(
  "type" = "s3",
  "AWS_ENDPOINT" = "http://127.0.0.1:15232",
  "AWS_REGION" = "region",
  "AWS_BUCKET" = "ws.test",
  "AWS_ROOT_PATH" = "cooldown",
  "AWS_ACCESS_KEY" = "ak",
  "AWS_SECRET_KEY" = "sk",
  "AWS_MAX_CONNECTIONS" = "50",
  "AWS_REQUEST_TIMEOUT_MS" = "3000",
  "AWS_CONNECTION_TIMEOUT_MS" = "1000"
)
```

```
CREATE STORAGE POLICY cooldown_policy
PROPERTIES
(
  "storage_resource" = "remote_s3",
  "cooldown_ttl" = "1d"
);
```



```
ALTER TABLE WH-WS
  set ("storage_policy" = "cooldown_policy")
```

## 其他特性应用

- TVF/TVF Schema。S3 数据文件 Schema 验证与数据导入；
- 基于 Nginx TCP 反向代理实现 FE 高可用；
- Unique 及 Aggregate 模型，其中 Aggregate 模型 用于大横表
- 用户资源配额
- Supervisor 自动拉起

...



# 4 展望

## 展望

### 有色工业数据平台

- 基于 **CCR** 实现总部-区域-生产单元多级数据复制
- 应用到其他业务板块，如矿山、新能源
- 升级至2.x，并紧跟 Doris 版本更新迭代

### For Doris

- JDK 11 +
- 继续加强安全性，如 SBOM
- 整合 AI



获取更多社区动态与最佳实践

## Apache Doris 官方平台:

- Apache Doris 官网: [doris.apache.org](https://doris.apache.org)
- Apache Doris GitHub: [github.com/apache/doris/](https://github.com/apache/doris/)

## 获取更多峰会资料:

- Doris Summit 峰会官网: [doris-summit.org.cn](https://doris-summit.org.cn)
- Doris Summit 峰会回放: <https://space.bilibili.com/1196172099/channel/collectiondetail?sid=1824324>