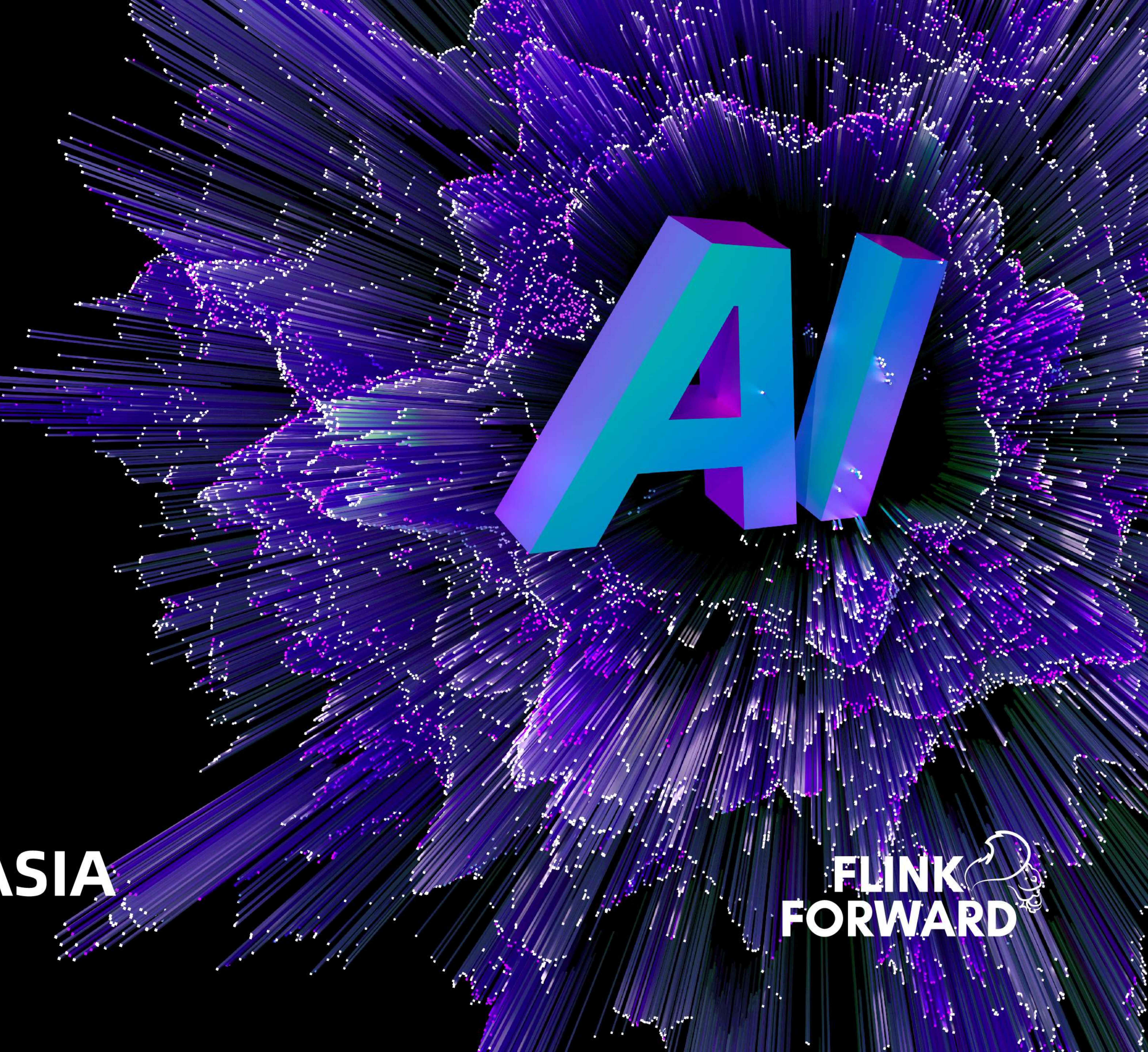
DLOnFlink

陈龙

Tencent Software Engineer

FLINK FORWARD # ASIA

实时即未来 # Real-time Is The Future



FLINK FORWARD

Contents
目录

01 项目介绍

What Is DLOnFlink?

02 架构设计

Challenges & Implementation Details.

03 未来发展与思考

Future Plan Of DLOnFlink

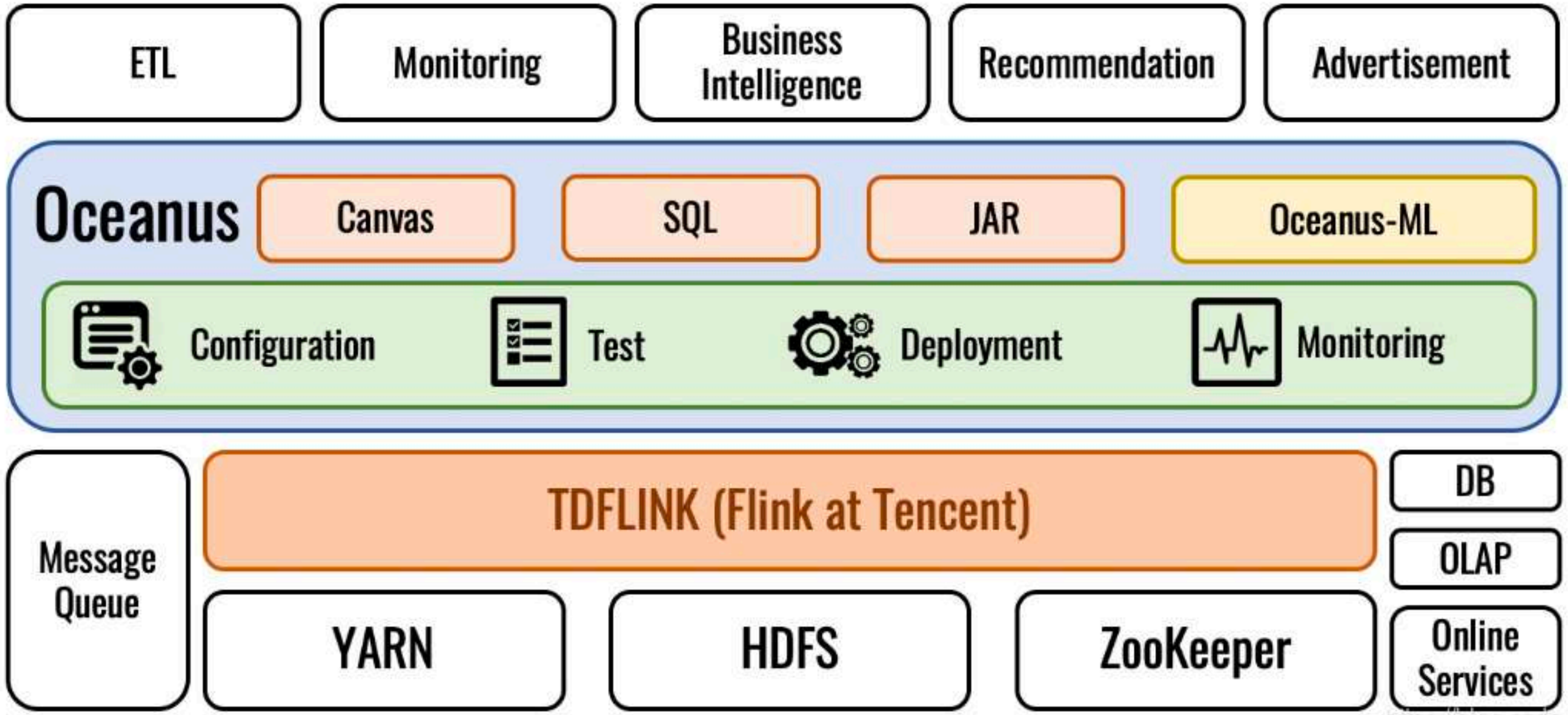
项目介绍

What Is DLOnFlink?

01

Oceanus Overview

A unified platform to develop and operate real-time applications



News

Advertisement

Business Intelligence

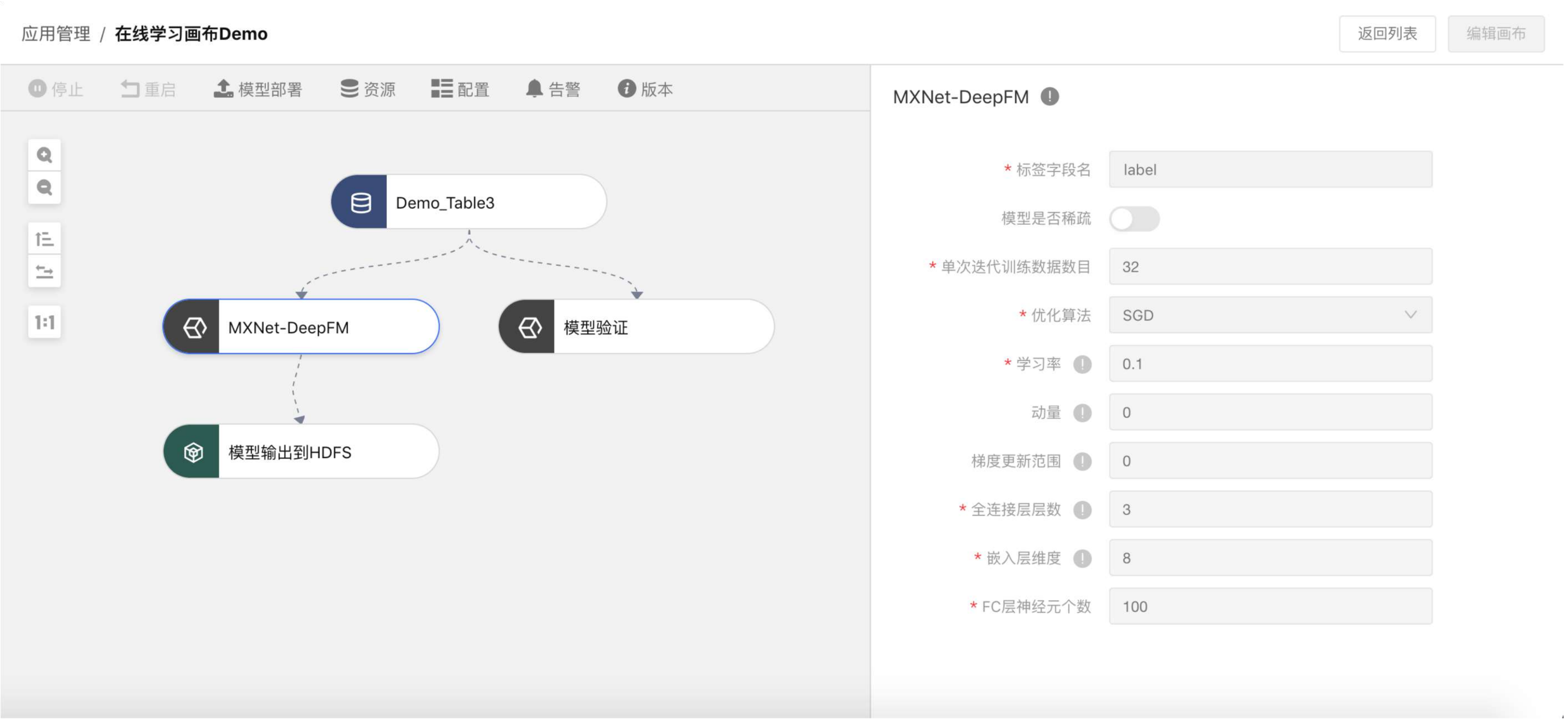
Games

210 Million
Maximum number of messages per second

500 PB
Amount of data

3 PB
Amount of data per day

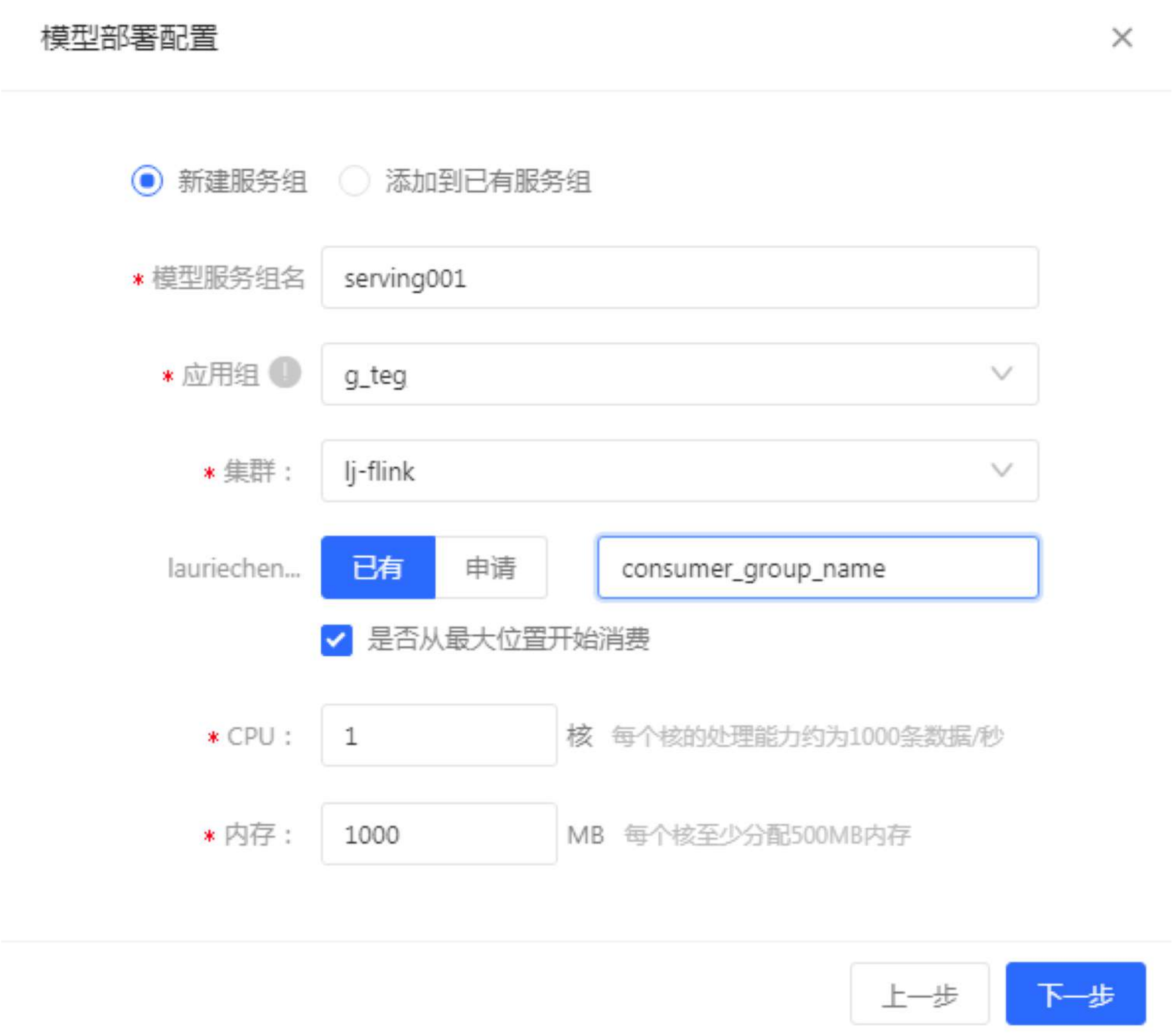
Oceanus-ML Demo



MXNet-DeepFM

* 标签字段名	label
模型是否稀疏	○
* 单次迭代训练数据数目	32
* 优化算法	SGD
* 学习率	0.1
动量	0
梯度更新范围	0
* 全连接层层数	3
* 嵌入层维度	8
* FC层神经元个数	100

Oceanus-ML Model Serving



模型部署配置 ×

● 新建服务组 ○ 添加到已有服务组

* 模型服务组名 serving001

* 应用组 g_teg

* 集群： lj-flink

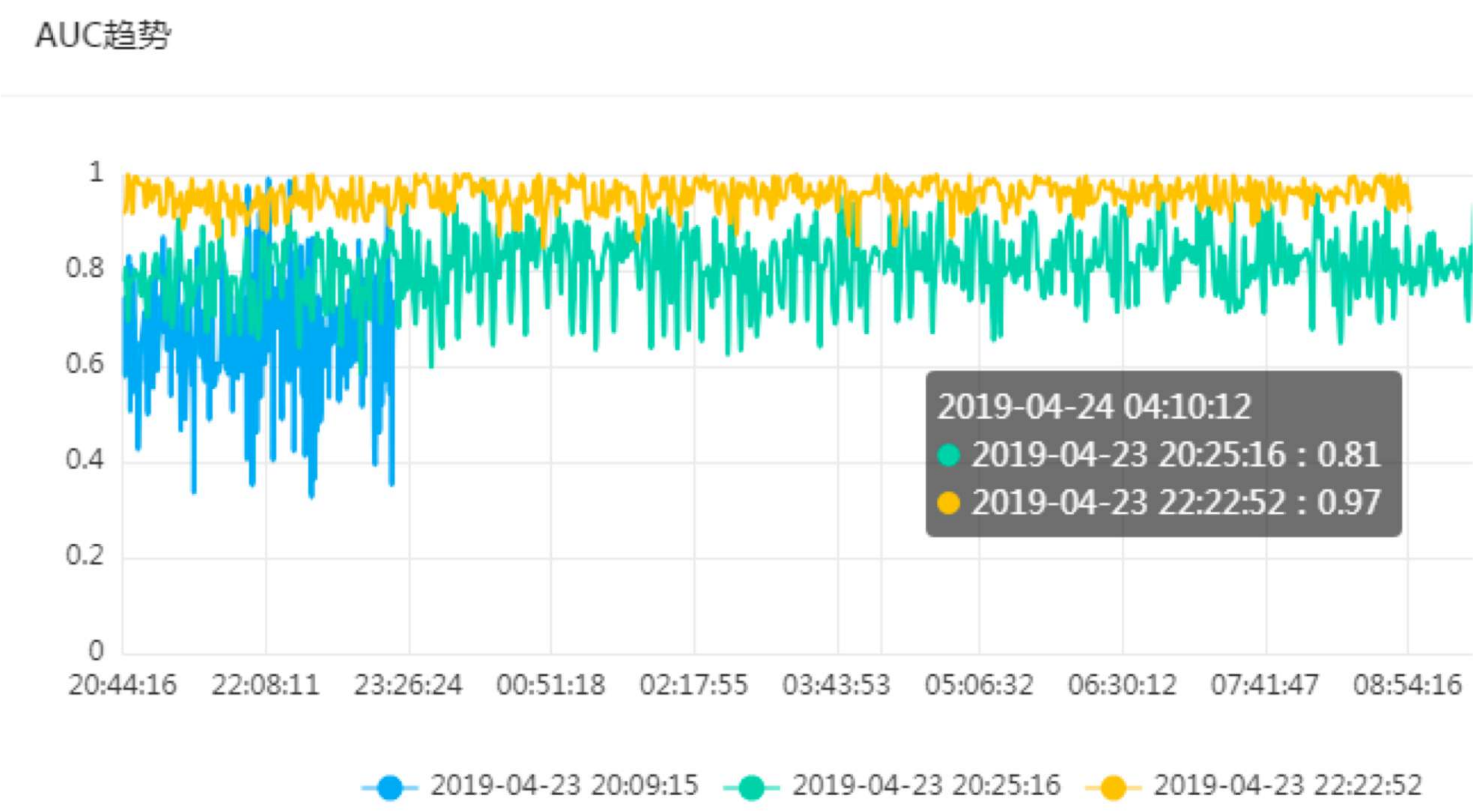
lauriechen... 已有 申请 consumer_group_name

☑ 是否从最大位置开始消费

* CPU： 1 核 每个核的处理能力约为1000条数据/秒

* 内存： 1000 MB 每个核至少分配500MB内存

Oceanus-ML Metrics



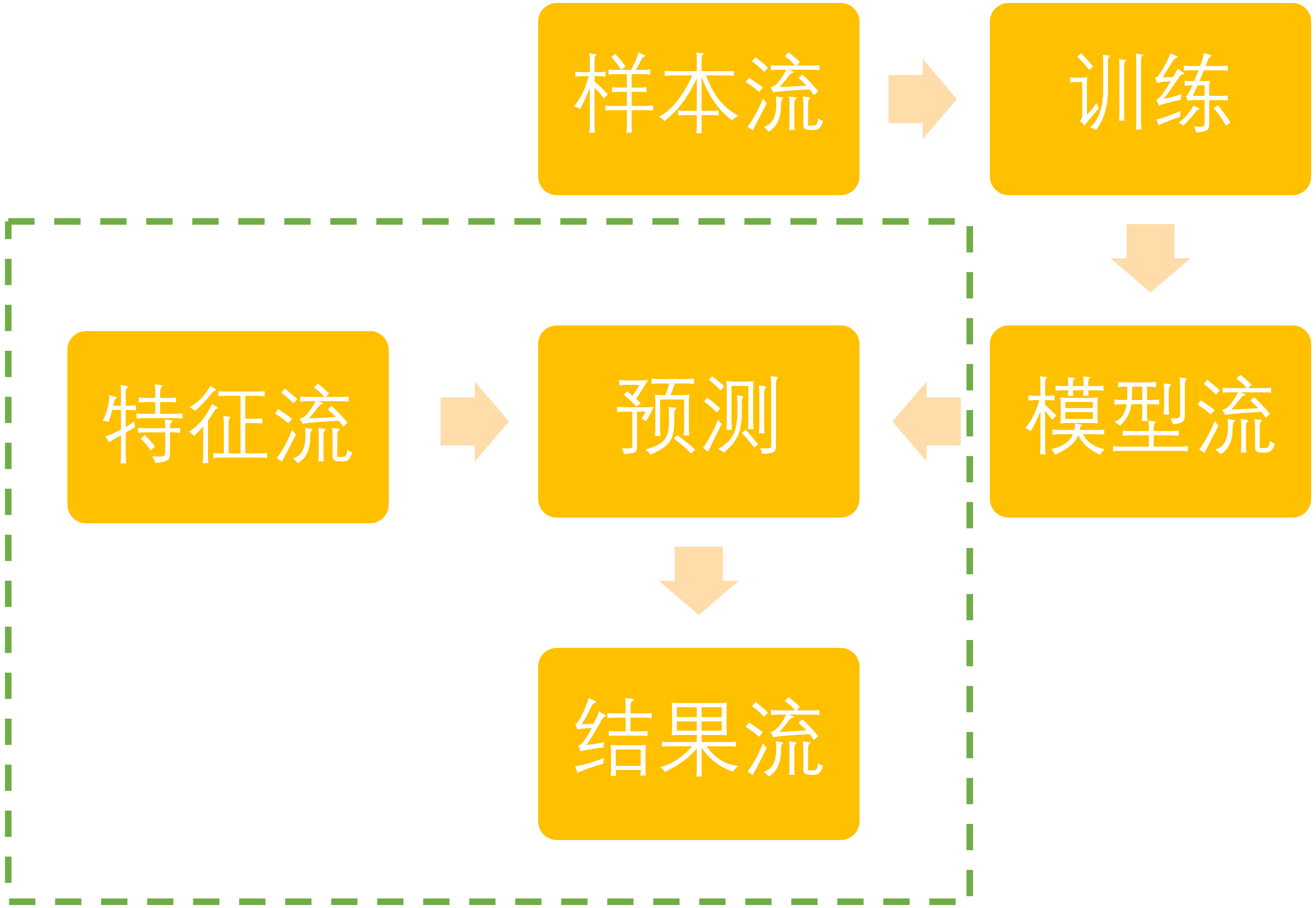
架构设计

Challenges & Implementation Details.

02

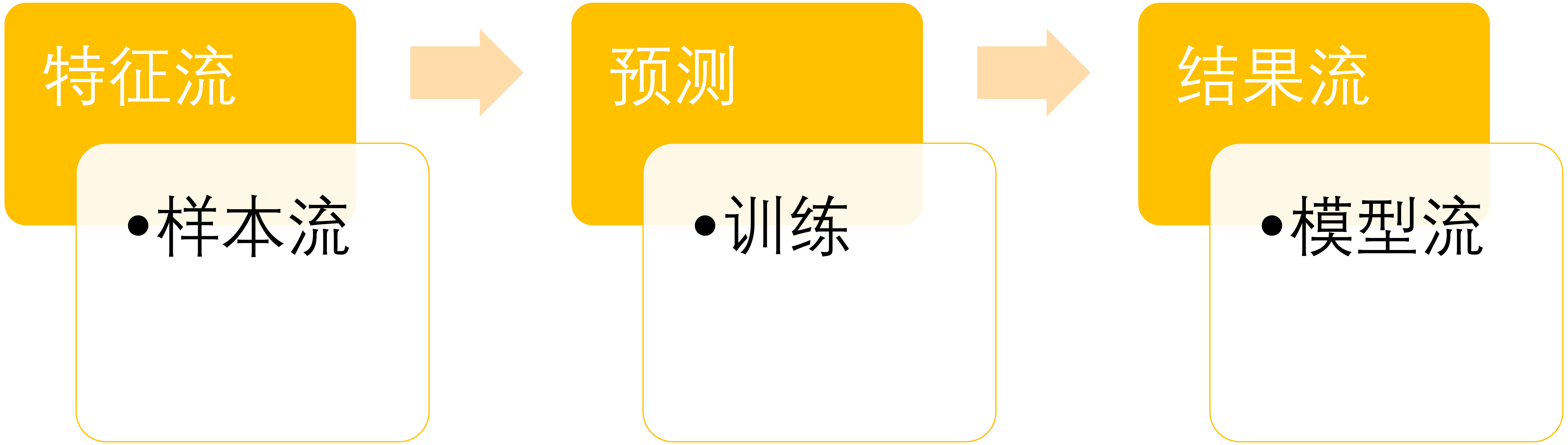
在线学习范式一

Online Learning Framework Ⅰ



在线学习范式二

Online Learning Framework Ⅱ



在线学习的出发点

Starting Point Of Online Learning

概念漂移

Concept drifting.

动态环境，如何处理反馈

How to deal with the feedback in the dynamic environment。

保证短期收益与增加多样性

Short term reward and diversity.

Deep Learning On Flink

结合Python端完备的ML生态，丰富的Model Zoo
With python ML framework and model zoo.

高效的计算图及自动求导
High efficient compute graph and autograd

方便整合离线生成的模型
Easily combine the offline learning model

框架上的改进

Improvement On Online Learning Framework

特征哈希解决动态增删ID

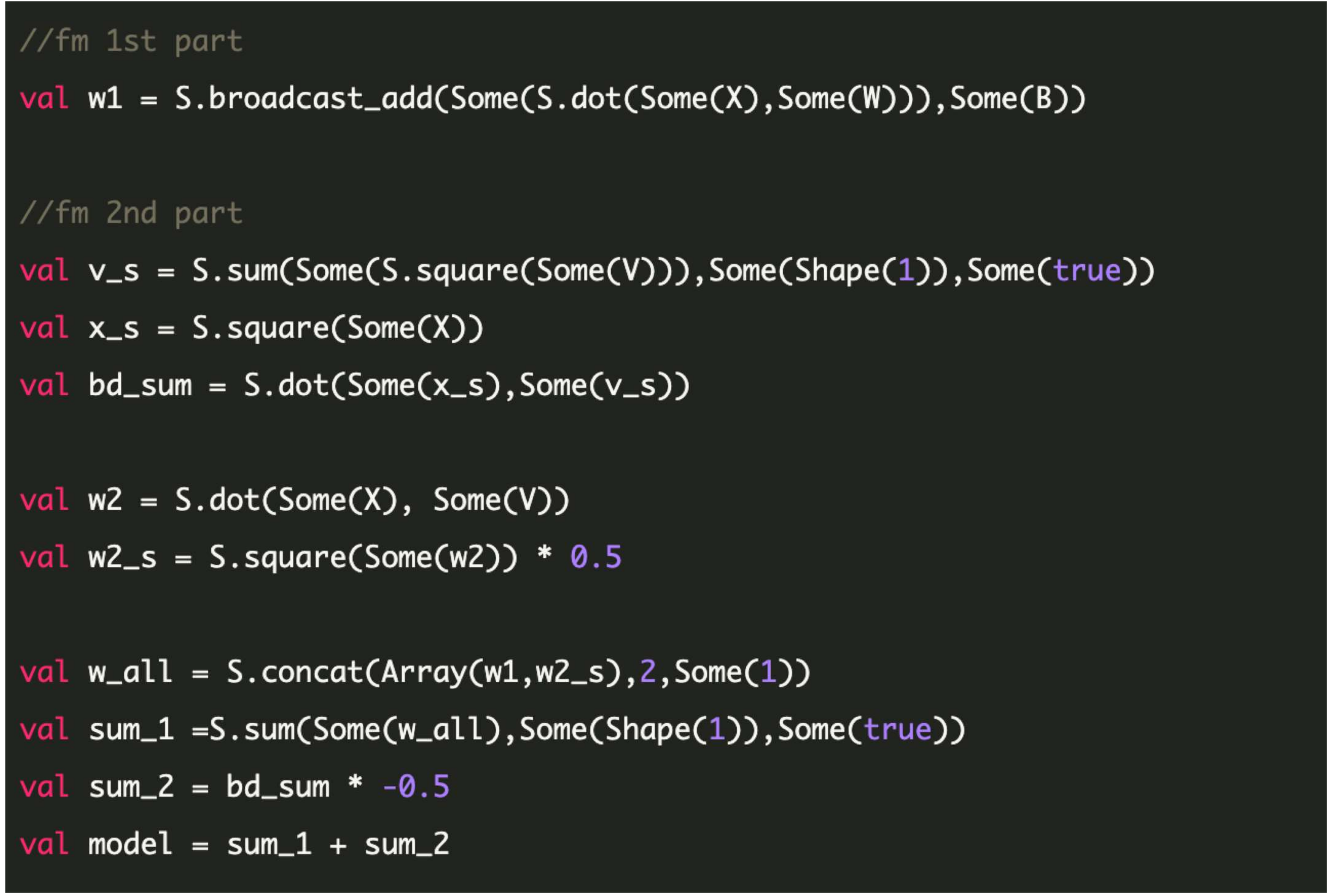
Dynamic add and delete ID feature by feature hashing

流式评估监控模型性能

Monitoring model performance continuous

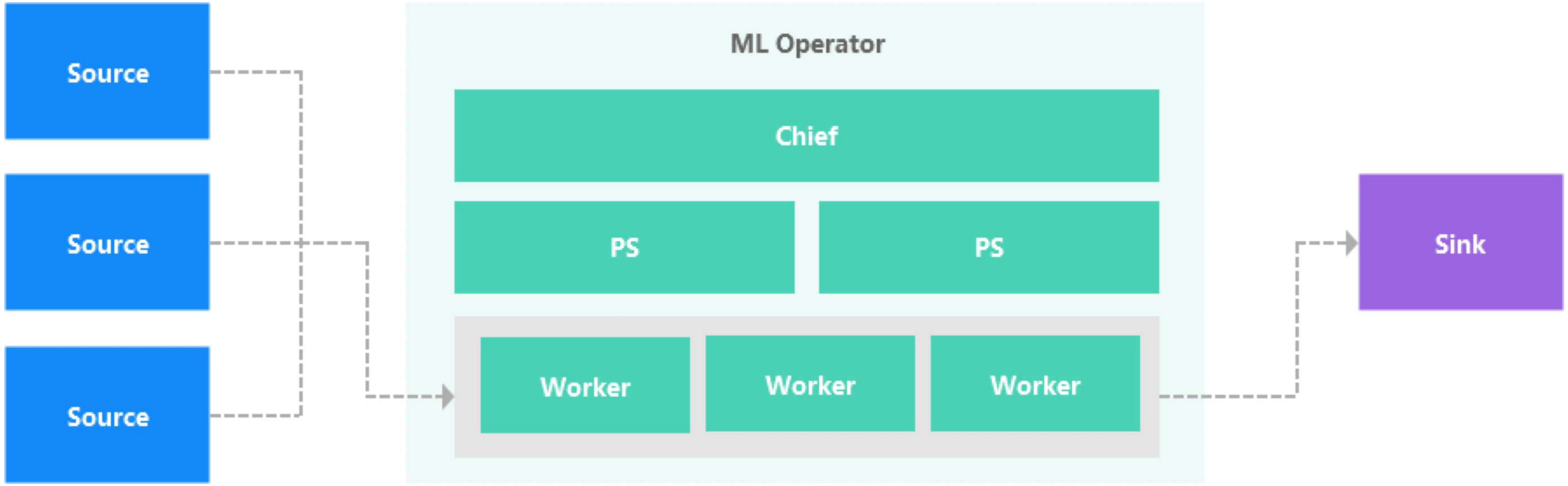
基于 MXNet 的 FM 算法

FM based on MXNet

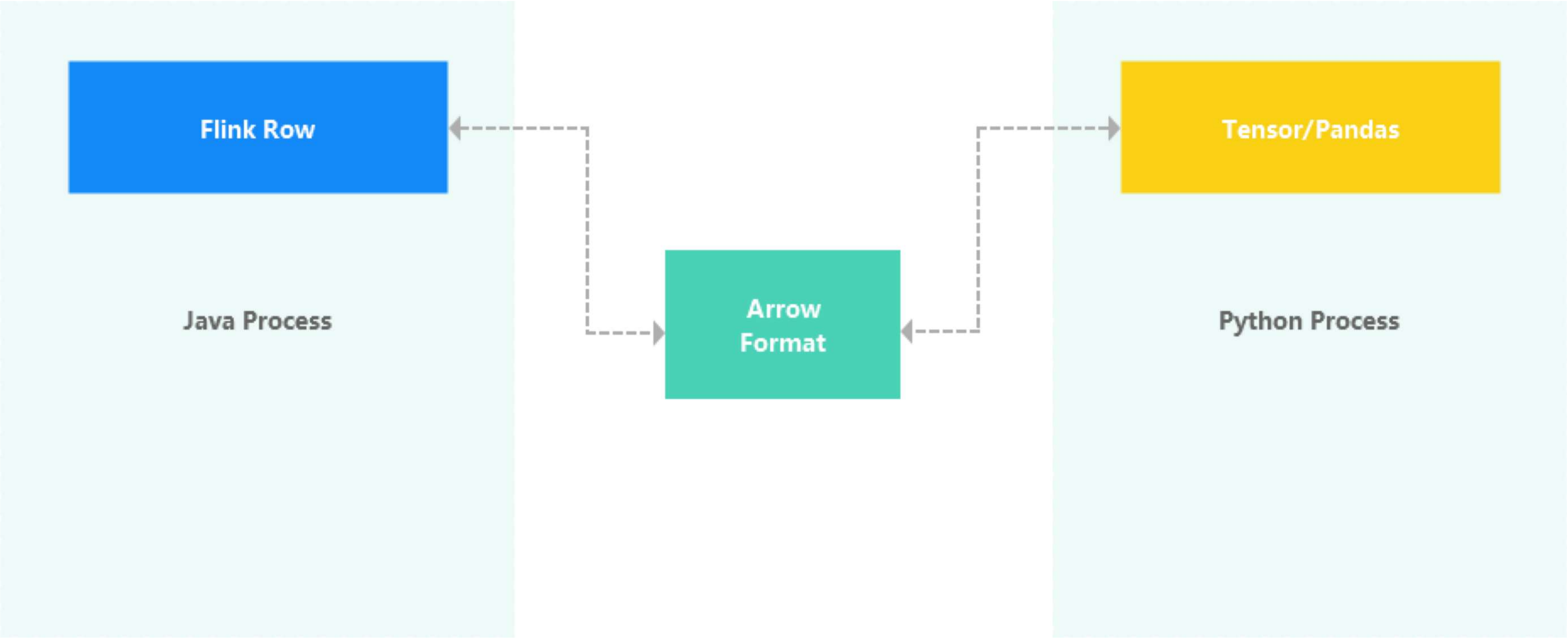


```
//fm 1st part
val w1 = S.broadcast_add(Some(S.dot(Some(X),Some(W))),Some(B))

//fm 2nd part
val v_s = S.sum(Some(S.square(Some(V))),Some(Shape(1)),Some(true))
val x_s = S.square(Some(X))
val bd_sum = S.dot(Some(x_s),Some(v_s))

val w2 = S.dot(Some(X), Some(V))
val w2_s = S.square(Some(w2)) * 0.5

val w_all = S.concat(Array(w1,w2_s),2,Some(1))
val sum_1 =S.sum(Some(w_all),Some(Shape(1)),Some(true))
val sum_2 = bd_sum * -0.5
val model = sum_1 + sum_2
```

DLOnFlink (Tensorflow， Pytorch)



DLOnFlink (Data Exchange)



Apache Arrow Format

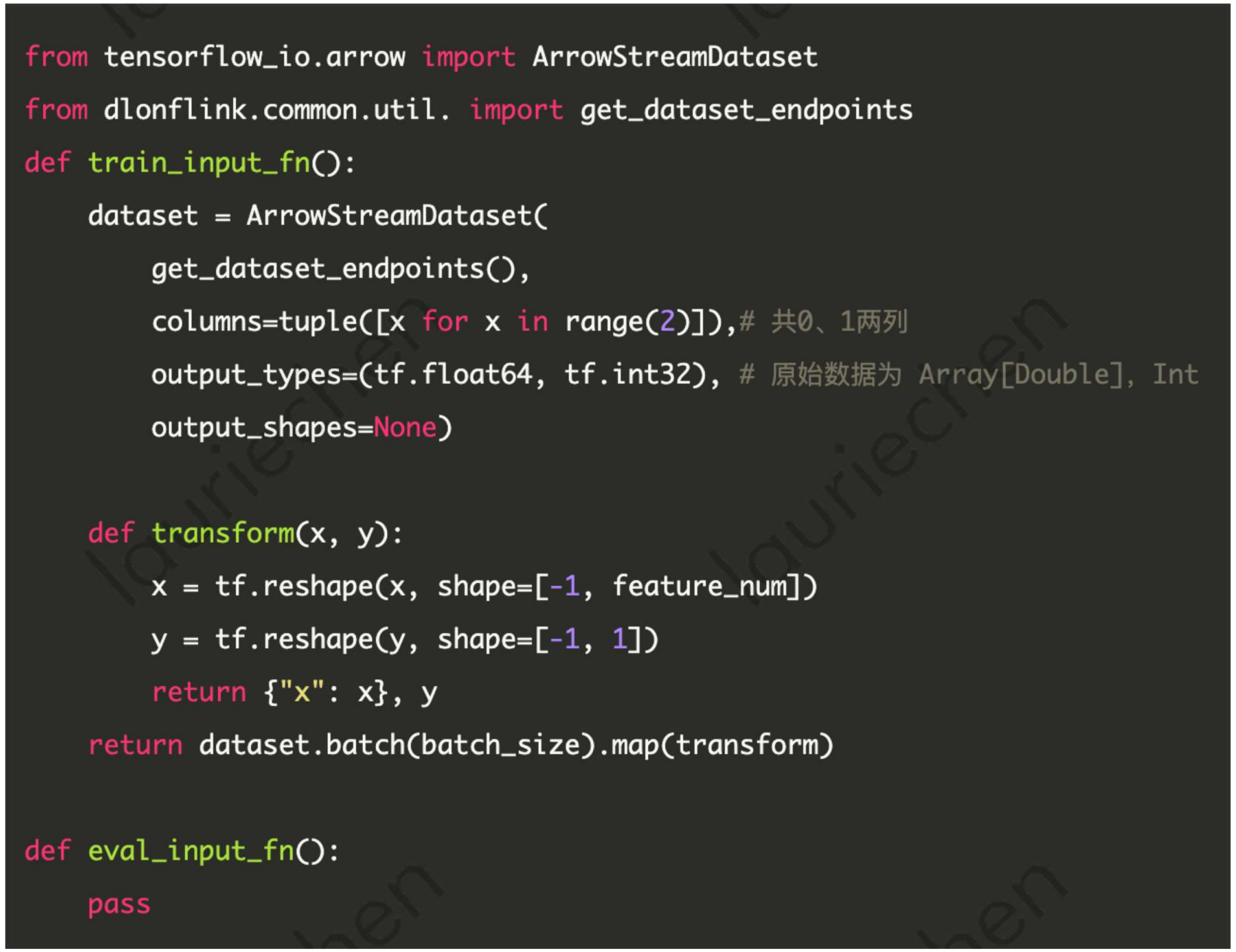
覆盖了 Flink 的基本类型（int，char， String, and Date ， etc。）及复合类型（数组，元组等）
Covering all types of Flink

零开销反序列化为pandas或者numpy
Zero cost deserialization to pandas or numpy

可以直接使用tensorflow io库提供的ArrowStreamReader读取处理
Easy to use with tensorflow io

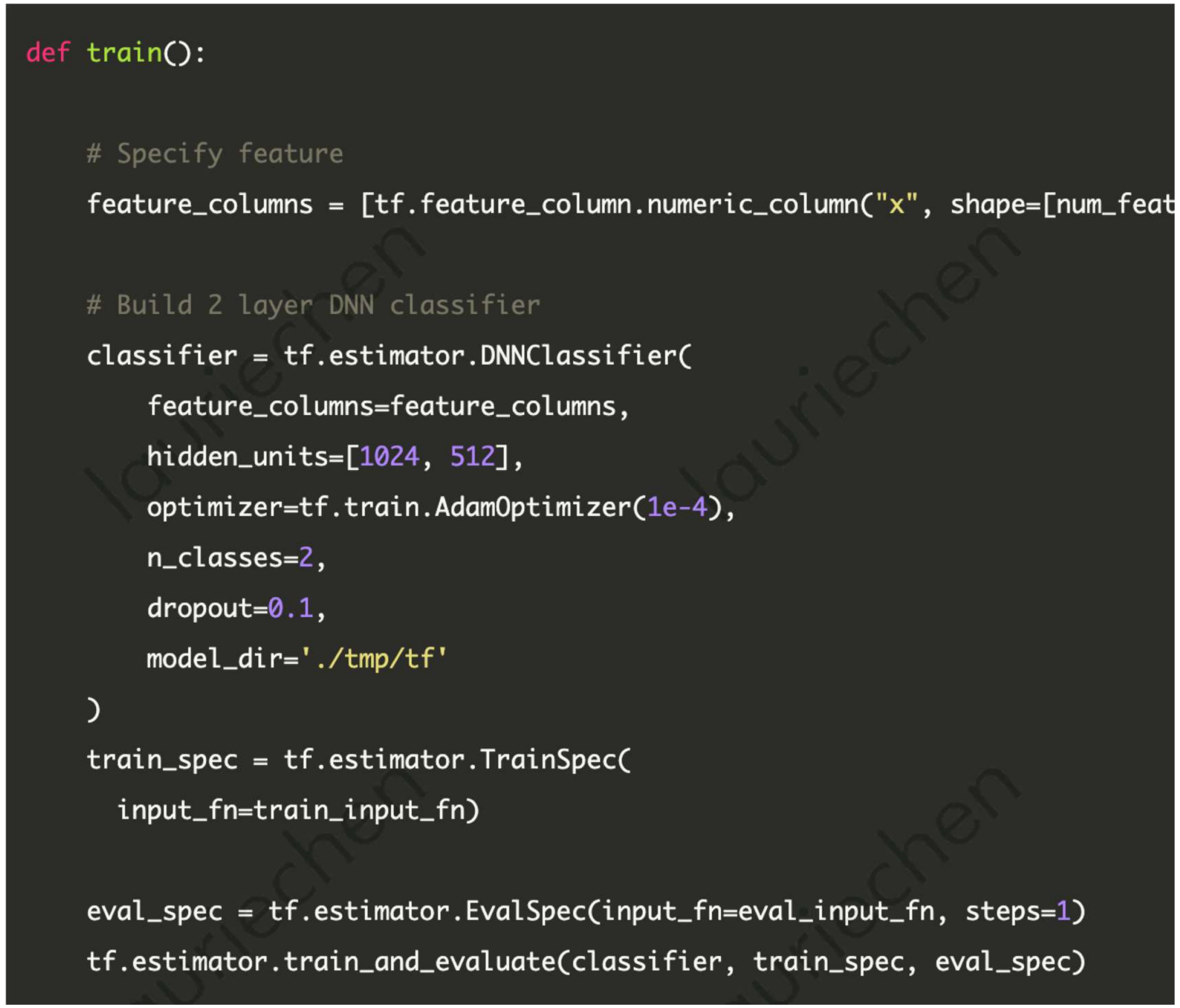
也可使用pyarrow库进行读写处理
Also support pyarrow

Tensorflow On Flink 示例代码（1/2）



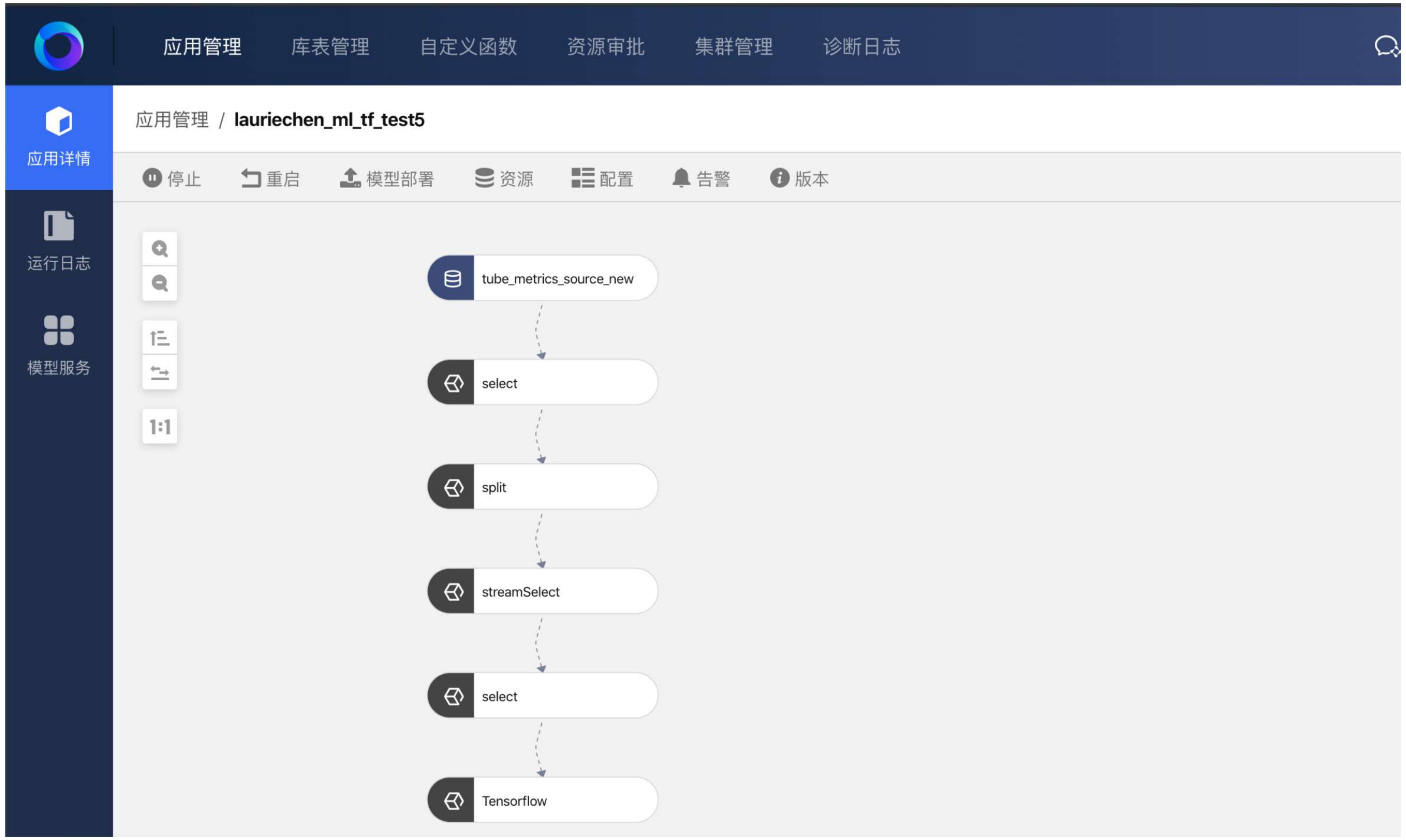
```
from tensorflow_io.arrow import ArrowStreamDataset
from dlonflink.common.util. import get_dataset_endpoints
def train_input_fn():
    dataset = ArrowStreamDataset(
        get_dataset_endpoints(),
        columns=tuple([x for x in range(2)]),# 共0、1两列
        output_types=(tf.float64, tf.int32), # 原始数据为 Array[Double]，Int
        output_shapes=None)

    def transform(x, y):
        x = tf.reshape(x, shape=[-1, feature_num])
        y = tf.reshape(y, shape=[-1, 1])
        return {"x": x}, y
    return dataset.batch(batch_size).map(transform)

def eval_input_fn():
    pass
```

Tensorflow On Flink 示例代码（2/2）

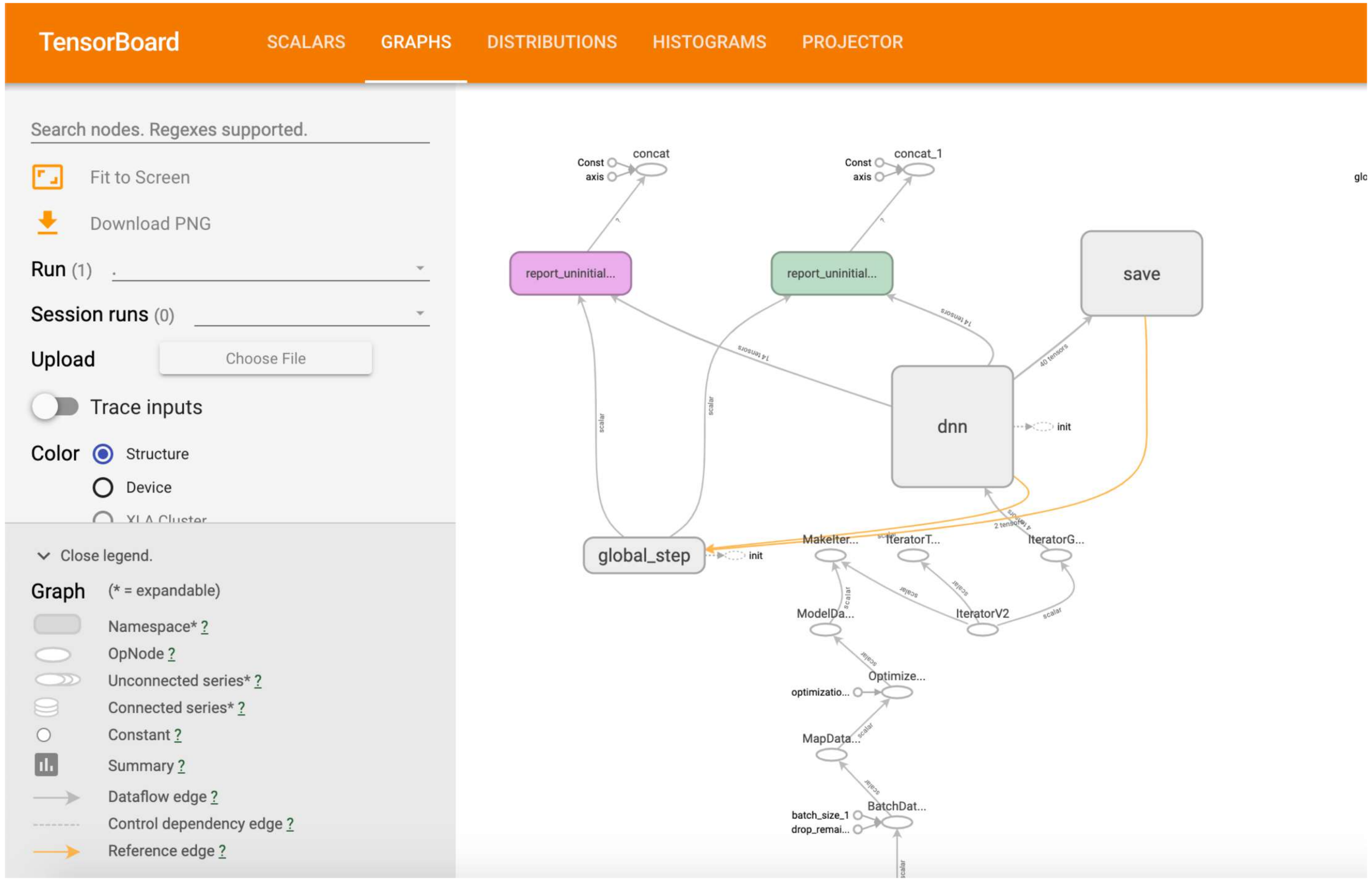


```
def train():

    # Specify feature
    feature_columns = [tf.feature_column.numeric_column("x", shape=[num_feat

    # Build 2 layer DNN classifier
    classifier = tf.estimator.DNNClassifier(
        feature_columns=feature_columns,
        hidden_units=[1024, 512],
        optimizer=tf.train.AdamOptimizer(1e-4),
        n_classes=2,
        dropout=0.1,
        model_dir='./tmp/tf'
    )
    train_spec = tf.estimator.TrainSpec(
      input_fn=train_input_fn)

    eval_spec = tf.estimator.EvalSpec(input_fn=eval_input_fn, steps=1)
    tf.estimator.train_and_evaluate(classifier, train_spec, eval_spec)
```

Tensorflow On Flink Pipeline



TensorBoard On Flink



Search nodes. Regexes supported.

Run (1) .

Session runs (0)

Upload

Trace inputs

Color ● Structure

○ Device

○ XLA Cluster

Close legend.

Graph (* = expandable)

Namespace* ?

OpNode ?

Unconnected series* ?

Connected series* ?

Constant ?

Summary ?

Dataflow edge ?

Control dependency edge ?

Reference edge ?

Contents
目录

01 项目介绍

What Is DLOnFlink?

02 架构设计

Challenges & Implementation Details.

03 未来发展与思考

Future Plan Of DLOnFlink

未来发展与思考

Future Plan Of DLOnFlink

03

未来规划

Future Plan Of DLOnFlink

增量保存模型

Save model incrementally

PS 作为服务常驻

Long time serving of parameter server

增加object store作为迭代缓存

Add object store for iteration

THANKS