



10101010101010101010101010
10101010101010101010
101010

OSC 源创会第 90 期 【在线直播】

AI 专场



直播时间：本周一至周四，每晚 20:00 - 21:00

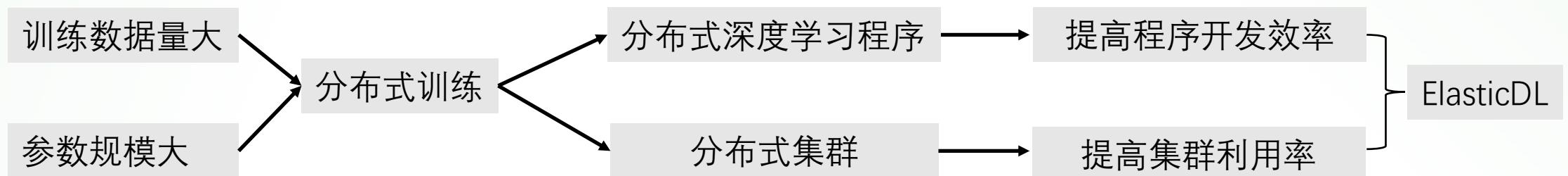


ElasticDL : Kubernetes-native 的弹性分布式深度学习框架

项目地址 <https://github.com/sql-machine-learning/elasticdl>

背景

在工业界生产场景中，深度学习应用具有训练数据量大和参数规模大的特点。



为了提供分布式程序开发效率和集群利用率，ElasticDL 提供了如下解决方案：

- 提供简单的分布式深度学习编程框架，让用户像写单机程序一样写分布式深度学习程序
- 弹性分布式深度学习系统，提高分布式作业的执行效率和集群资源利用率

背景

ElasticDL 是基于 Kubernetes 和 TensorFlow 2.x 实现的分布式深度学习训练系统。

使用 TensorFlow/Keras API 开发深度学习程序



Kubernetes 集群上运行分布式程序



kubernetes

- TensorFlow 是业界应用最广泛的深度学习框架
- end-to-end 的深度学习平台
- eager execution 大幅提升了TensorFlow 的易用性

- 目前最先进的分布式操作系统
- 公有云和私有云的事实工业标准
- Docker 容器隔离

ElasticDL : 像写单机程序一样写分布式深度学习程序

ElasticDL : Kubernetes-native 弹性分布式训练系统

ElasticDL 在蚂蚁集团 CTR 预估场景的实践

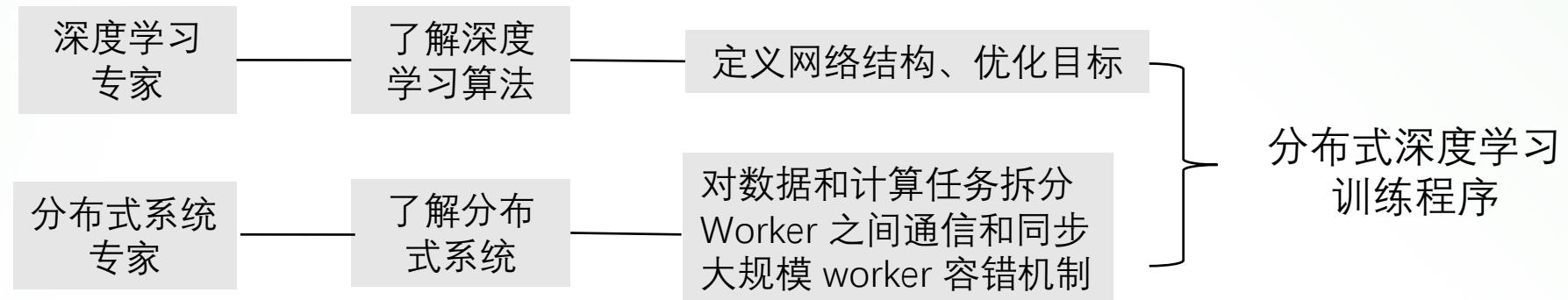
➤ ElasticDL：像写单机程序一样写分布式深度学习程序

ElasticDL：Kubernetes-native 弹性分布式训练系统

ElasticDL 在蚂蚁集团 CTR 预估场景的实践

ElasticDL：像写单机程序一样写分布式深度学习程序

分布式深度学习程序难写的原因：



Kubernetes 上运行 TensorFlow 分布式训练程序的一些开源解决方案：

	模型定义	Kubernetes 调度工具	缺点
方案一	TensorFlow Estimator API	Kubeflow TF-operator	仅支持 graph execution, 不支持 eager execution
方案二	TensorFlow Keras API	Kubeflow TF-operator	不支持 Parameter Server 分布式
方案三	Horovod with TensorFlow	Kubeflow MPI-operator	了解 TensorFlow 和 Horovod API

使用 Kubeflow 提供的 Kubernetes Operator 在集群上运行分布式训练作业，还需要用户对 Kubernetes 有所了解。

ElasticDL：像写单机程序一样写分布式深度学习程序

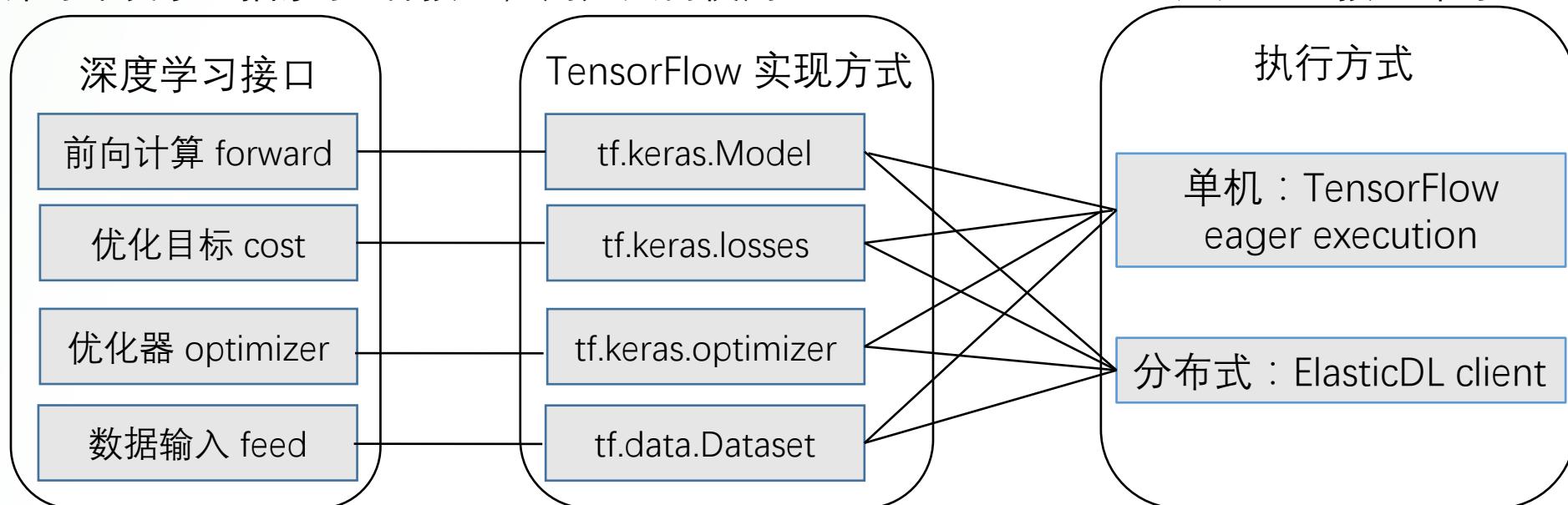
- ElasticDL 使用 TensorFlow eager execution 开发了分布式的 training loop，用户无需关心分布式系统。

```
w = tf.Variable([[1.0]])
with tf.GradientTape() as tape:
    loss = w * w

grad = tape.gradient(loss, w)
print(grad) # => tf.Tensor([[ 2.]], shape=(1, 1), dtype=float32)
```

tf.Tensor([[2.]], shape=(1, 1), dtype=float32)

- 针对深度学习抽象了4种接口，用户只需使用 TensorFlow/Keras API 定义这些接口即可。



ElasticDL 模型开发：https://github.com/sql-machine-learning/elasticdl/blob/develop/docs/tutorials/model_building.md

ElasticDL：像写单机程序一样写分布式深度学习程序

mnist 数据集手写数字识别模型

```
def forward():
    inputs = tf.keras.Input(shape=(28, 28), name="image")
    x = tf.keras.layers.Reshape((28, 28, 1))(inputs)
    x = tf.keras.layers.Conv2D(32, kernel_size=(3, 3), activation="relu")(x)
    x = tf.keras.layers.Conv2D(64, kernel_size=(3, 3), activation="relu")(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.MaxPooling2D(pool_size=(2, 2))(x)
    x = tf.keras.layers.Dropout(0.25)(x)
    x = tf.keras.layers.Flatten()(x)
    outputs = tf.keras.layers.Dense(10)(x)
    return tf.keras.Model(inputs=inputs, outputs=outputs, name="mnist_model")

def loss(labels, predictions):
    labels = tf.reshape(labels, [-1])
    return tf.reduce_mean(
        tf.nn.sparse_softmax_cross_entropy_with_logits(
            logits=predictions, labels=labels
        )
    )

def optimizer(lr=0.1):
    return tf.optimizers.SGD(lr)

def dataset_fn(dataset, mode, _):
    dataset = dataset.map(_parse_data)

    if mode == Mode.TRAINING:
        dataset = dataset.shuffle(buffer_size=1024)
    return dataset
```

生成模型镜像

```
FROM tensorflow
RUN pip install elasticdl
COPY model_zoo /model_zoo
```

ElasticDL client 提交分布式训练

```
elasticdl train \
--image_name=elasticdl:ci \
--model_zoo=model_zoo \
--model_def=mnist_functional_api.mnist_functional_api.custom_model \
--training_data=/data/mnist/train \
--num_epochs=1 \
--master_resource_request="cpu=1, memory=4096Mi, ephemeral-storage=1024Mi" \
--worker_resource_request="cpu=1, memory=4096Mi, ephemeral-storage=1024Mi" \
--ps_resource_request="cpu=1, memory=4096Mi, ephemeral-storage=1024Mi" \
--minibatch_size=64 \
--num_ps_pods=1 \
--num_workers=2 \
--job_name=test-train \
--distribution_strategy=ParameterServerStrategy \
--output=model_output
```

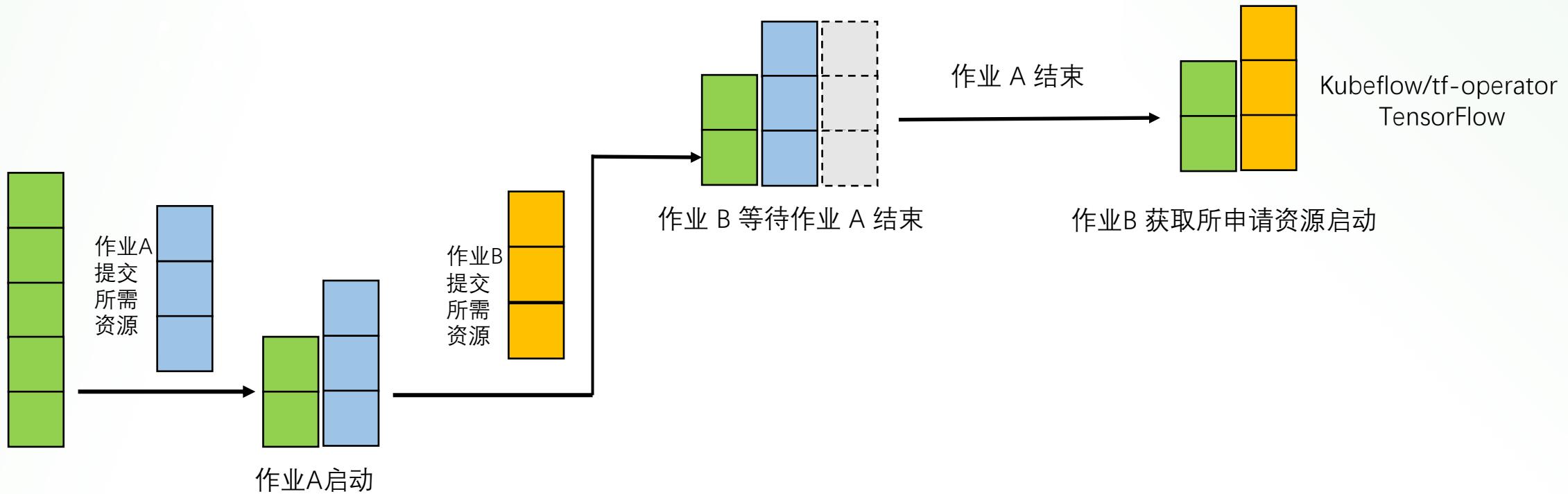
ElasticDL : 像写单机程序一样写分布式深度学习程序

➤ ElasticDL : Kubernetes-native 弹性分布式训练系统

ElasticDL 在蚂蚁金服 CTR 预估场景的实践

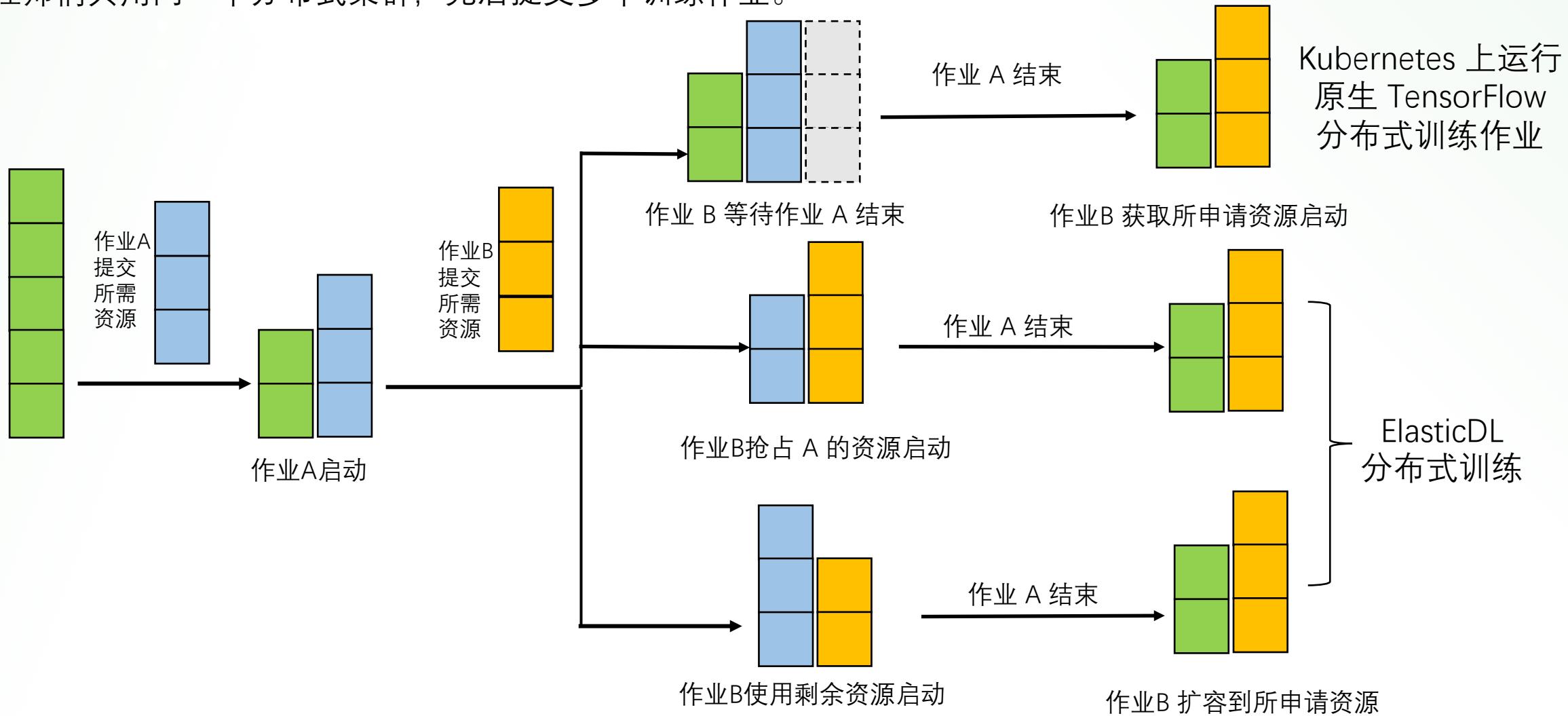
共用集群资源，资源等待时间长，资源利用率低

工程师们共用同一个分布式集群，先后提交多个训练作业。



共用集群资源，资源等待时间长，资源利用率低

工程师们共用同一个分布式集群，先后提交多个训练作业。



- 调用 TensorFlow eager execution 构建分布式训练系统
- 调用 Kubernetes API 来调整训练过程中的进程数量
- 数据分片的动态分发机制

ElasticDL: 调用 TensorFlow eager execution 构建分布式训练系统

基于 TensorFlow 的分布式训练系统大致分为以下四类，ElasticDL 位于田字格的右下角是为了容错和弹性。

	TensorFlow 1.x graph mode	TensorFlow 2.x eager execution
in TensorFlow runtime	TensorFlow's parameter server	TensorFlow distributed strategy
above TensorFlow API	Horovod	ElasticDL, Horovod

TensorFlow runtime: 与平台无关，不会调用集群管理系统的来调整训练资源，无法实现主动的弹性调度。

TensorFlow API : ElasticDL 和 Horovod 都是基于 TensorFlow API 来实现分布式训练。

相同点：

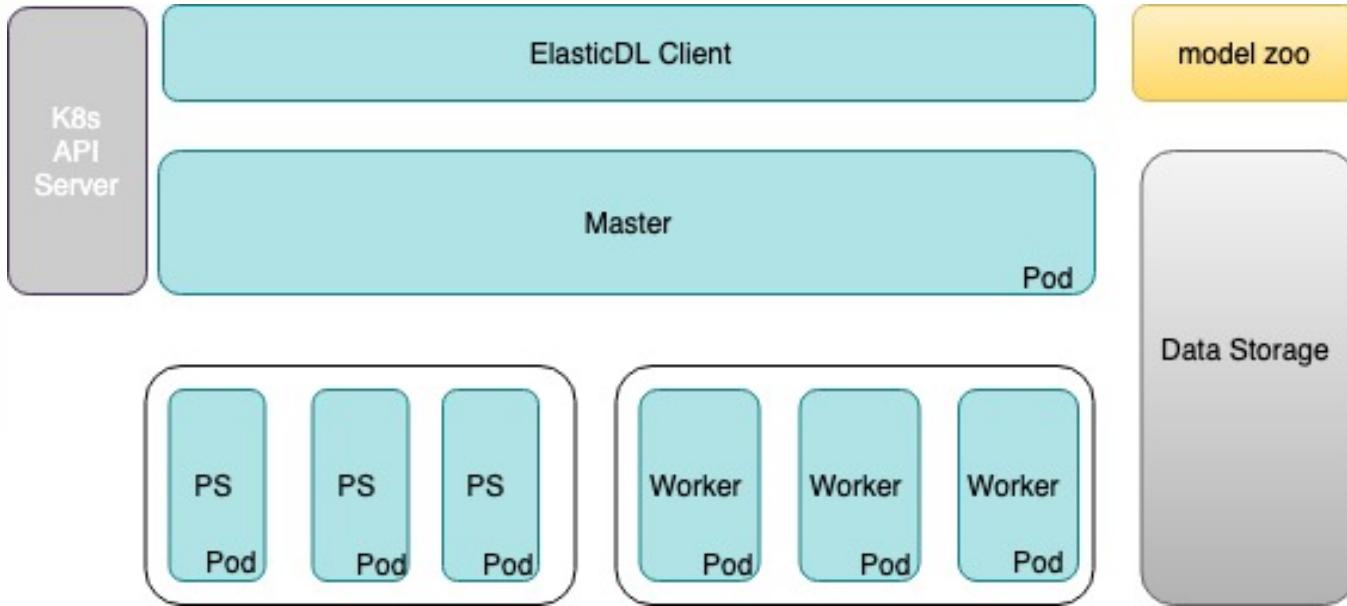
- 每个进程有完成的计算逻辑
- 通过 TensorFlow API 获取梯度

不同点：

- Horovod 与平台无关，可以运行在多种分布式集群上，但是不能利用集群管理系统来调整进程数量进行弹性调度
- ElasticDL 是 Kubernetes-native，只能运行在 Kubernetes 集群上，能利用 Kubernetes API 来调整作业进程数量

ElasticDL : 调用 Kubernetes API 来调整训练过程中的进程数量

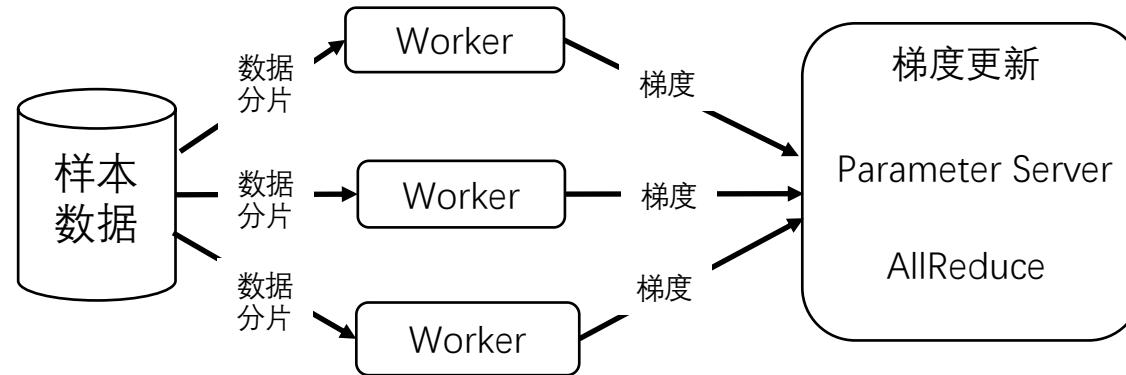
ElasticDL 通过在 Kubernetes 上创建 master 进程来协调训练数据分配、通信、梯度同步和容错，从而支持弹性调度。



- ElasticDL Client 通过 Kubernetes API 启动 Master 进程
- Master 通过 Kubernetes API 启动 PS 和 Worker 进程
- Master 给 worker 分发数据分片
- Master 通过 Kubernetes API 来监听 Worker 状态
- Master 通过 Kubernetes API 重启被抢占的 Worker

ElasticDL : 数据分片的动态分发机制

分布式训练作业启动多个 worker 进程后，需要给 worker 进程分配数据分片。



为了提高分布式训练的效率，数据分片要尽可能均匀，防止 worker 间的负载不均衡。

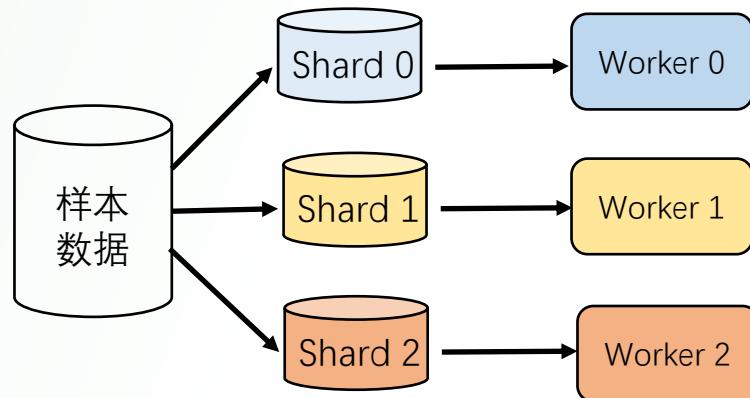
为了达到弹性调度，数据分片还需满足如下条件：

- Worker 失败后，其计算的数据分片能分配给其他 Worker 重新计算
- 有新的 Worker 加入后，能获取为计算的数据分片进行计算

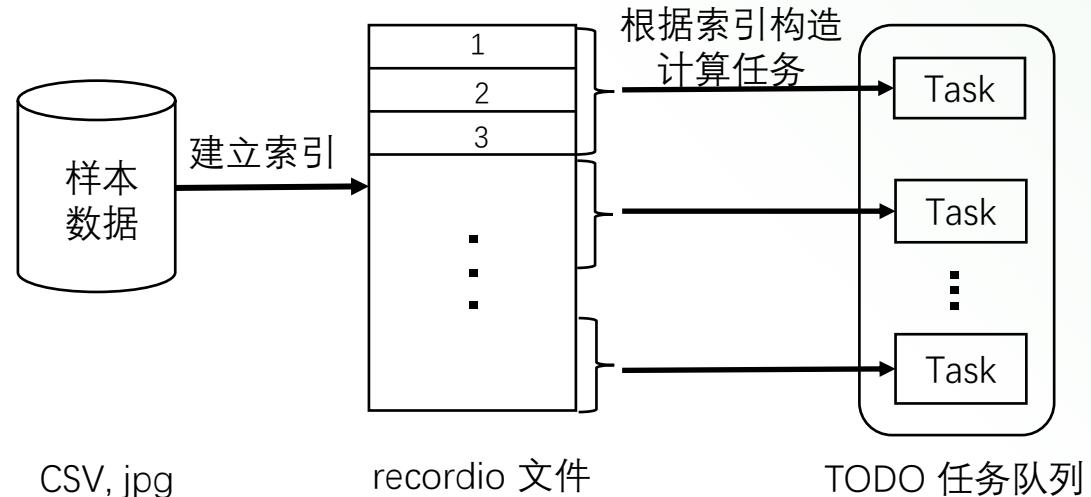
ElasticDL : 数据分片的动态分发机制

静态数据分发

在作业开始前，将数据进行分片，然后指定给某个 Worker

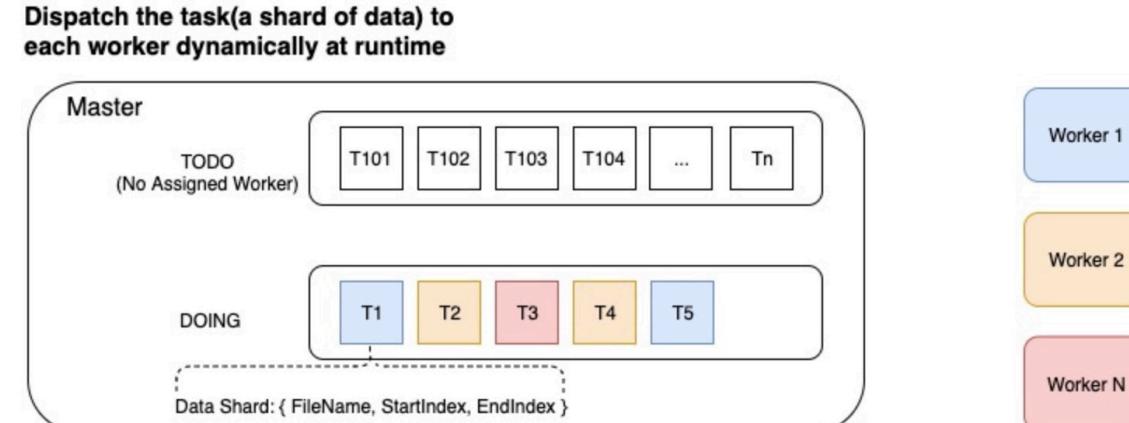


ElasticDL 动态数据分发



缺点：

- Worker 失败后，其分配的数据没法给其他 Worker 训练
- 慢 Worker 会拖慢整个训练作业
- 数据分片遍历次数不均衡，影响收敛性

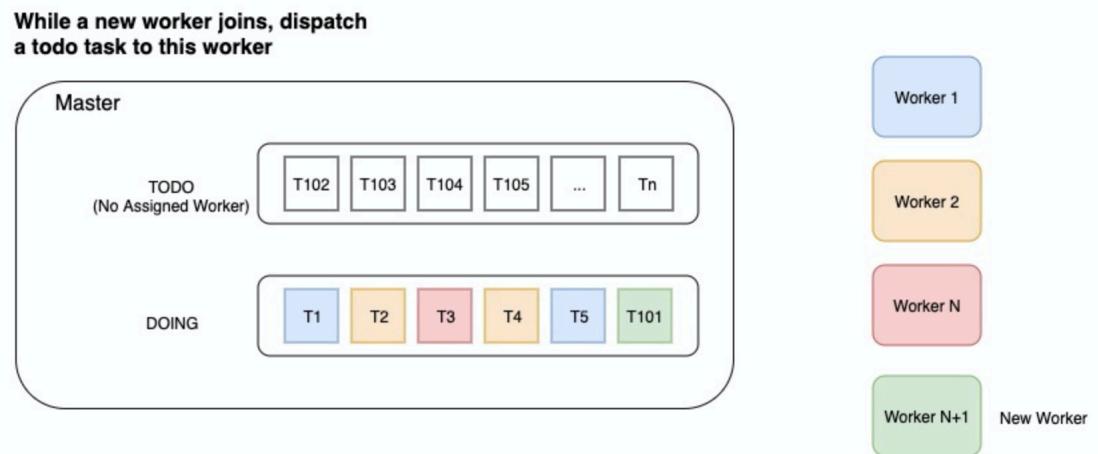


ElasticDL : 数据分片的动态分发机制

ElasticDL : 调整进程数量时的数据分发

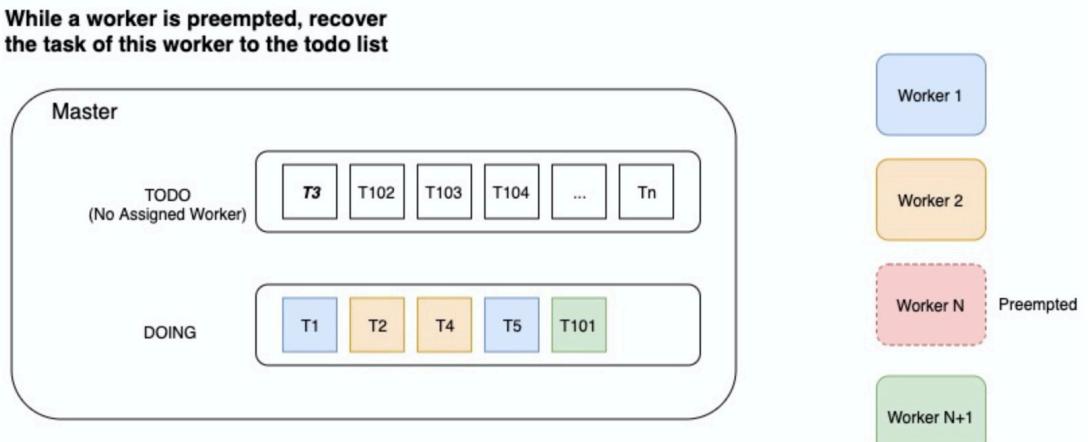
新的 Worker 加入作业 (扩容)

Master 将新的 Worker 加入作业时，会从 TODO队列里给新 Worker 分配 Task，Worker 根据 Task 拿到数据分片进行计算



Worker 被其他作业抢占 (缩容)

- Master 回收被抢占 Worker 的 Task 到 TODO队列
- Master 将 Task 分配给其他正常 Worker 计算



为了验证 ElasticDL 弹性调度的能够提高深度学习作业的研发效率和集群资源利用率，分别做了三组实验：

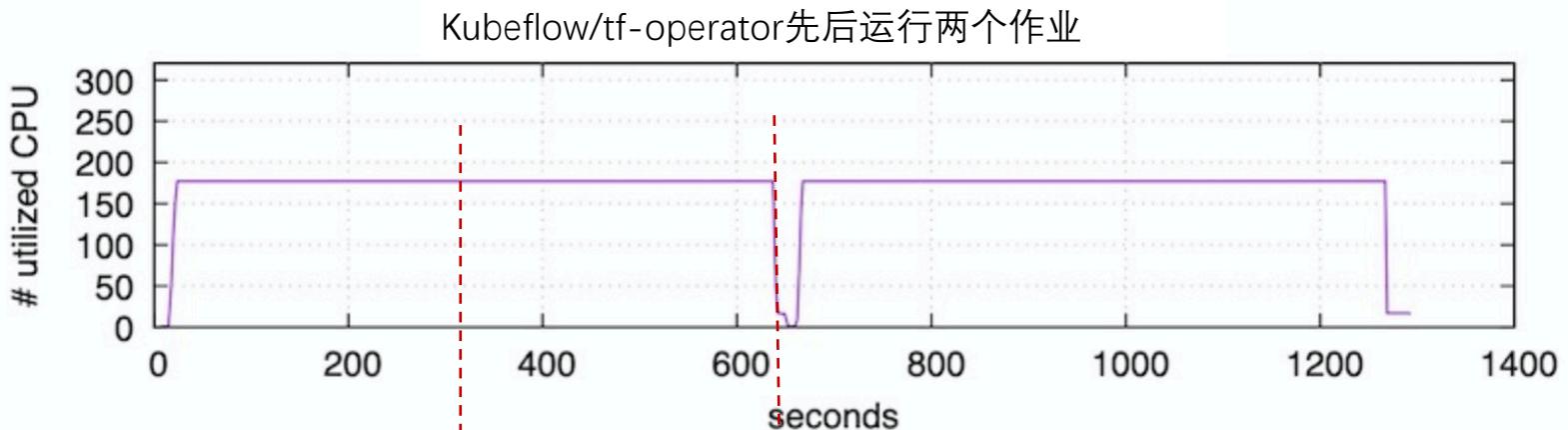
- 多个深度学习训练作业同时在集群上启动，验证弹性调度能缩短作业等待时间
- 深度学习作业与高优先级的在线服务混布，验证弹性调度可以提升集群资源利用率
- 训练时调整 worker 数量观察模型收敛性，验证 ElasticDL 弹性调度不影响模型收敛

ElasticDL benchmark 实验——多个深度学习训练作业同时在集群上启动

两个深度学习训练作业需要的资源总和略超过集群的情况

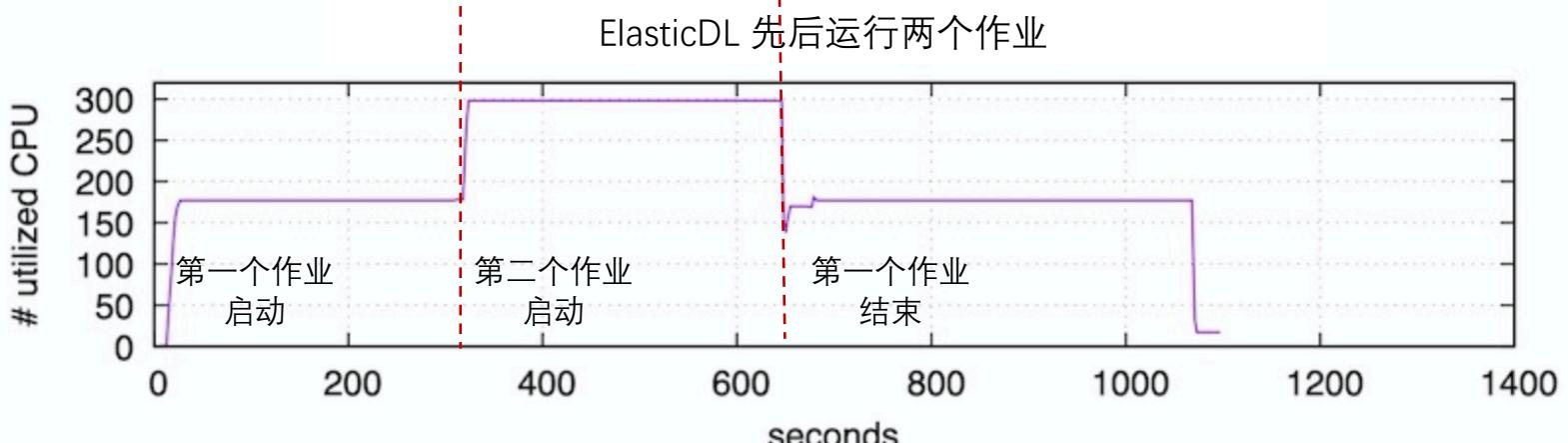
实验设置：

- 集群总 CPU 数 320 个
- 先后提交两个训练作业
- 每个作业需要 175 个CPU



实验结论：

- 用户作业等待时间几乎为 0
- 集群利用率高
- 作业完成时间更快



ElasticDL benchmark 实验——深度学习作业与高优先级在线服务混布

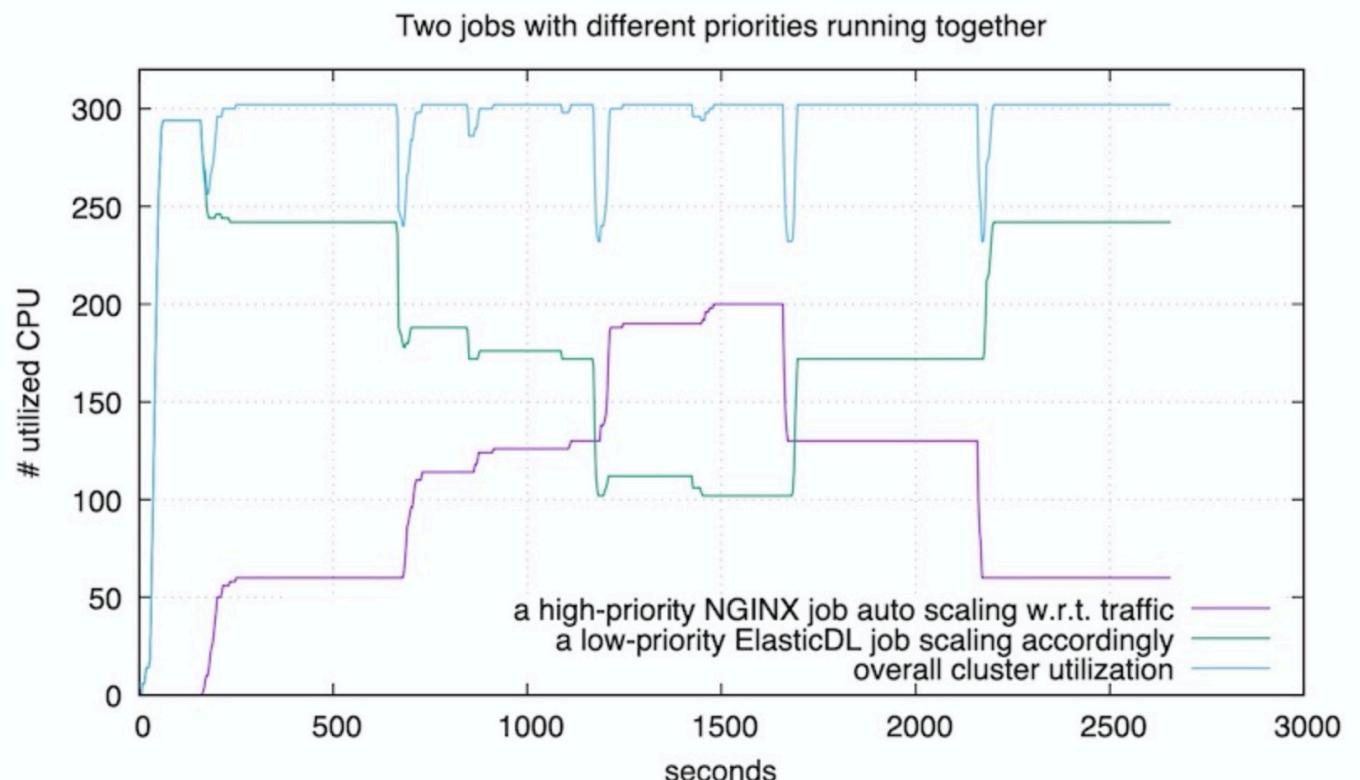
ElasticDL 作业与在线 nginx 服务运行在同一个 Kubernetes 集群上，nginx 作业的优先级高于ElasticDL作业。

实验设置：

- 集群总 CPU 数 320 个
- 训练任务 DeepFM 二分类模型
- 训练时长约 40 min

实验结论：

- 当用户请求增加时，Kubernetes 自动扩容 nginx 服务，ElasticDL 作业释放部分资源
- 流量高峰过去后，nginx 服务释放资源，ElasticDL 扩充资源



ElasticDL benchmark 实验——训练时调整 worker 数量不影响模型收敛性

训练过程中，ElasticDL 调整的 worker 数量，不会超过用户配置的最大 worker 数。防止 worker 数量过多，梯度更新的参数滞后严重，影响模型收敛性。

实验数据集：

Kaggle Display Advertising Challenge

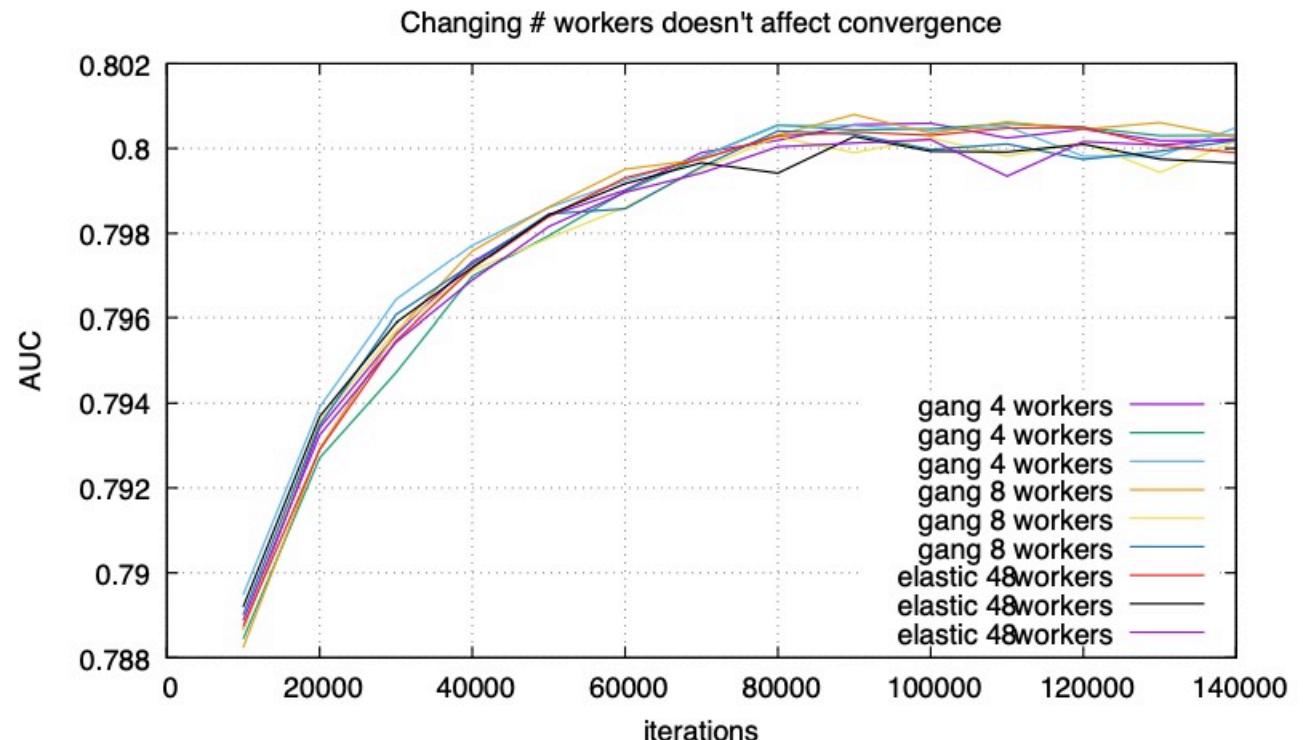
数据列	描述
Label 列	1 或 0, 1 表示点击, 0 表示未点击
I1-I13	整数型特征
C1-C26	字符串型类别特征

样本数量：约4千万

模型：Wide & Deep Learning

参数更新策略：Parameter Server 下的异步 SGD

实验结论：弹性调度的模型 AUC 和 kubeflow/tf-operator 提交的原生 TensorFlow 分布式训练作业持平。



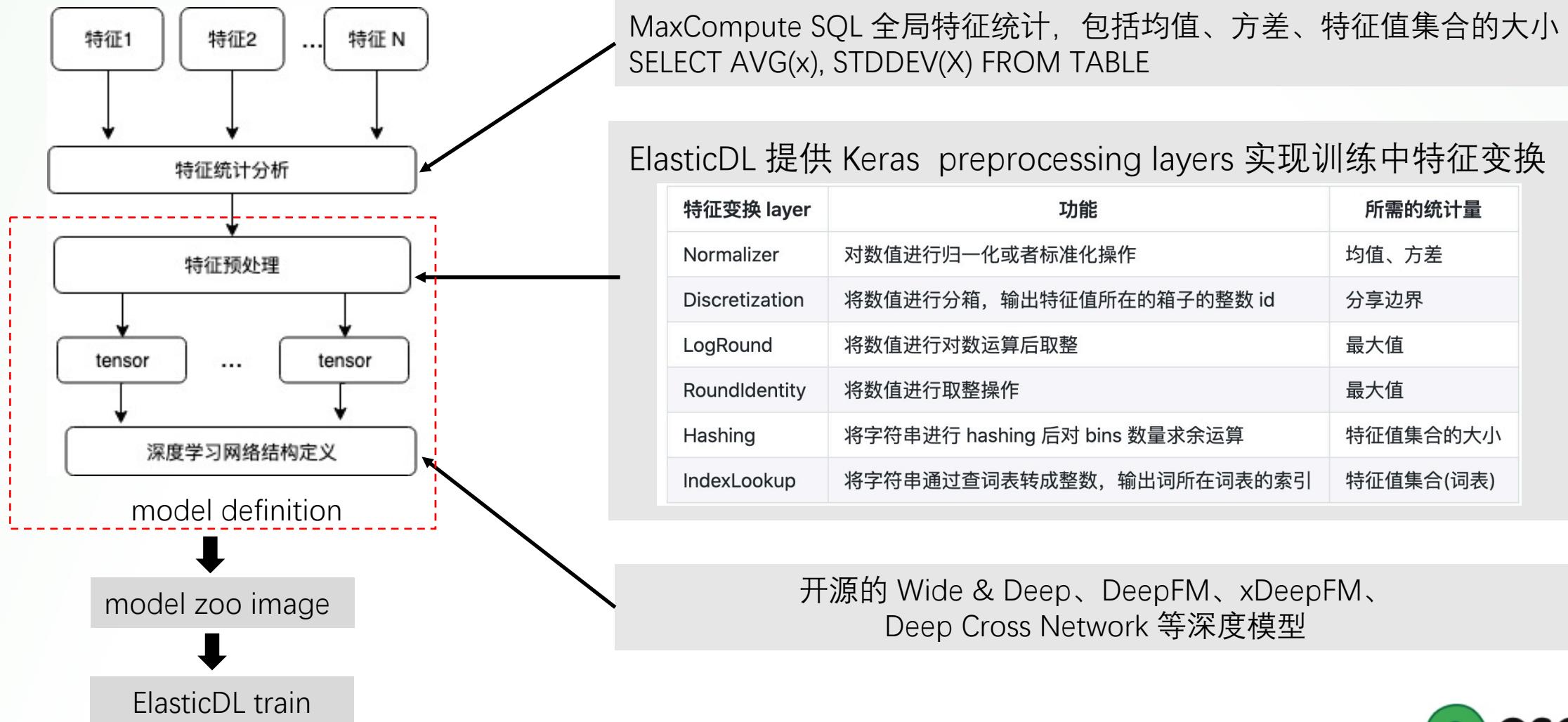
ElasticDL : 像写单机程序一样写分布式深度学习程序

ElasticDL : Kubernetes-native 弹性分布式训练系统

➤ ElasticDL 在蚂蚁集团 CTR 预估场景的实践

ElasticDL 在蚂蚁集团 CTR 预估场景的实践

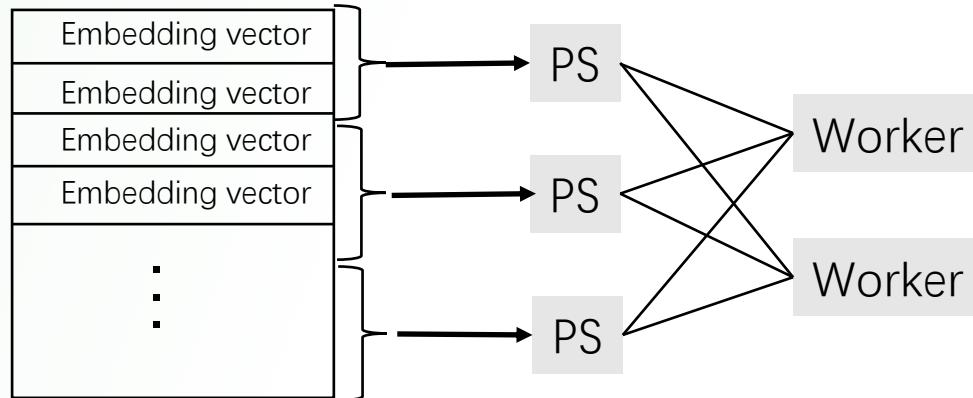
蚂蚁集团的搜索推荐广告场景应用到了大量的 CTR 预估模型。其数据以结构化的形式存储在阿里云的 MaxCompute 上。CTR 预估模型建模流程如下：



ElasticDL 在蚂蚁集团 CTR 预估场景的实践

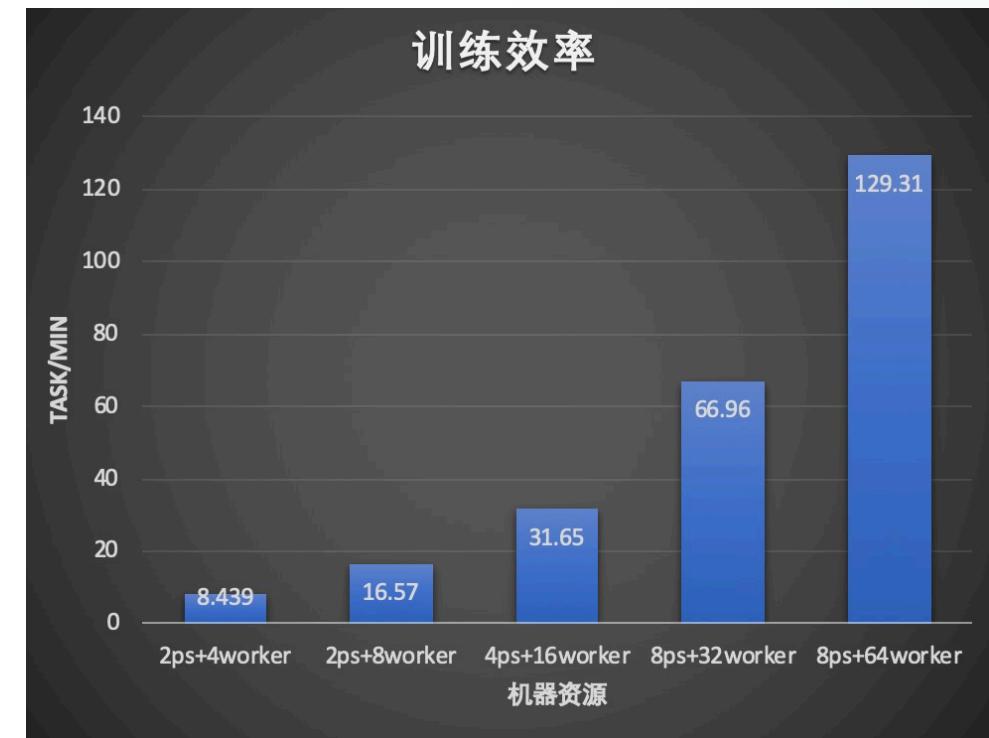
CTR 预估模型的输入一般含有用户和商品属性的高维稀疏特征，需要使用大规模的 embedding，通常采用 Parameter Server (PS) 来训练这种模型。ElasticDL 针对 PS 的优化：

ElasticDL Embedding 的优化



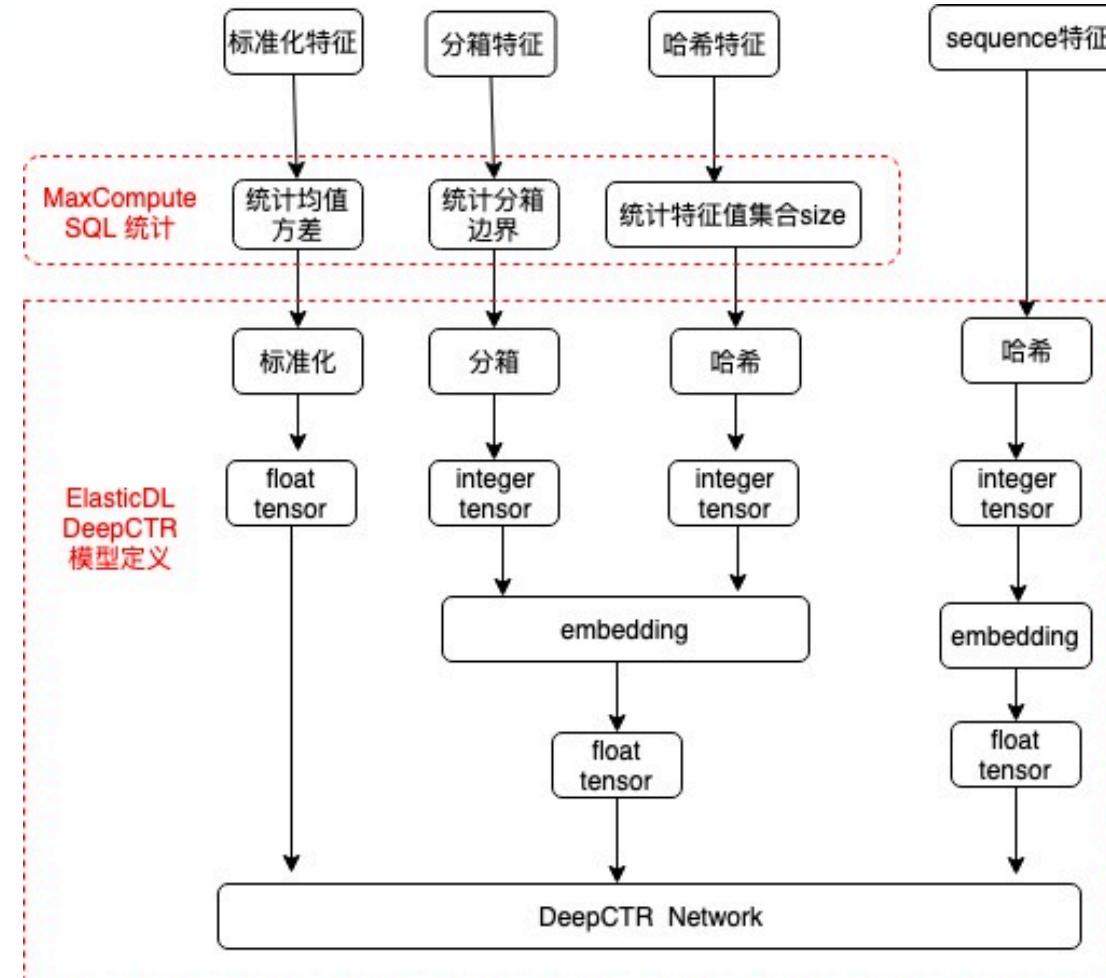
- Embedding vector 惰性初始化，无需提前指定 embedding 的大小；
- Embedding 表分拆到多个 PS 上存储和更新，均衡存储和通信负载；
- Worker 从 PS 请求参数和发送梯度时，针对重复 ID 做合并，减少通信量。

PS 并发度优化：训练速率与 worker 数量线性相关



ElasticDL 在蚂蚁集团 CTR 预估场景的实践

为了进一步提升易用性，ElasticDL 结合 DeepCTR 预估模型，开发了自动特征预处理的建模方案，用户只需配置特征处理方式即可完成 CTR 模型的分布式训练：



ElasticDL 在蚂蚁集团 CTR 预估场景的实践

为了验证方案的性能，选用 Kaggle Display Advertising Challenge 数据集进行测试。

样本数量：训练集约 4000 万条，测试集约 600 万条

label	i1	i2	i3	i4	i5	i6	i7	i8	i9	i10	i11	i12	i13	c1	c2	c3	c4	c5	c6
0	1	1	5	0	1382	4	15	2	181	1	2	\N	2	68fd1e64	80e26c9b	fb936136	7b4723c4	25c83c98	7e0ccccf
1	\N	47	\N	0	6399	38	19	10	143	\N	10	\N	6	1464facd	38a947a1	223b0e16	ca55061c	25c83c98	7e0ccccf
0	\N	38	2	4	3119	149	64	48	139	\N	6	6	4	05db9164	26a88120	d032c263	c18be181	4cf72387	fbad5c96
0	\N	56	2	\N	\N	\N	0	0	1	\N	0	\N	\N	5a9ed9b0	8084ee93	02cf9876	c18be181	25c83c98	fbad5c96
0	\N	0	\N	\N	14919	\N	0	0	0	\N	0	\N	\N	05db9164	4c2bc594	d032c263	c18be181	384874ce	fe6b92e5
1	\N	-1	\N	\N	5760	42	218	0	287	\N	15	0	\N	5a9ed9b0	3f0d3f28	\N	\N	25c83c98	7e0ccccf
0	0	16	44	21	1844	77	5	21	70	0	1	\N	21	5a9ed9b0	4f25e98b	606b404a	ecb0d7f8	4cf72387	7e0ccccf
1	7	31	24	6	7	7	4	6	1	1	\N	6	05db9164	09e68b86	aa8c1539	85dd697c	25c83c98	7e0ccccf	
0	3	26	1	1	1431	1	11	1	1	0	1	\N	1	68fd1e64	c76014f5	df00c4f0	031bba14	25c83c98	\N
0	0	0	6	13	1509	116	6	47	222	0	2	1	61	5bfa8ab5	b7ca2abd	ee96fc95	68ad052c	25c83c98	7e0ccccf
0	\N	1	23	3	72413	3607	0	4	44	\N	0	\N	3	68fd1e64	f0cf0024	74e1a23a	9a6888fb	25c83c98	fbad5c96
0	\N	4	6	2	9205	64	26	6	242	\N	5	0	2	b455c6d7	bce95927	edde3156	13508380	4cf72387	7e0ccccf
0	1	0	\N	9	27	9	1	10	9	1	1	\N	9	68fd1e64	d83353f5	77f2f2e5	d16679b9	384874ce	fe6b92e5
0	\N	164	7	2	10118	259	20	11	256	\N	5	0	30	39af2607	26a88120	b00d1501	d16679b9	25c83c98	fbad5c96
0	2	0	1	\N	188	0	2	0	0	1	1	\N	05db9164	e5fb1af3	3783c552	2a85af5	25c83c98	7e0ccccf	
1	0	0	2	8	1632	56	16	37	56	0	4	0	8	68fd1e64	09e68b86	aa8c1539	85dd697c	25c83c98	fbad5c96
0	\N	62	18	6	90	\N	0	6	6	\N	0	\N	6	05db9164	80e26c9b	4d88f82c	169ffff5	43b19349	\N
1	0	2	1	19	2914	451	6	49	107	0	1	1	19	05db9164	3df44d94	d032c263	c18be181	0942e0a7	fbad5c96
0	\N	-1	\N	\N	\N	0	0	6	\N	0	\N	0	05db9164	b961056b	d1b59691	8eb681c0	25c83c98	\N	
0	0	2	92	12	2993	238	4	30	376	0	2	\N	16	05db9164	287130e0	b367491d	d46c3907	25c83c98	7e0ccccf
0	\N	0	1	15	381	306	0	26	221	\N	0	\N	23	8cf07265	ae46a29d	c5deb4f6	f922efad	25c83c98	7e0ccccf
0	\N	-1	\N	\N	4374	5	8	0	52	\N	1	0	\N	05db9164	86d4fcc	61610e17	a754f0ee	4cf72387	fe6b92e5
0	1	165	8	1	2	0	2	2	10	1	2	\N	0	41edad3d	09e68b86	4a224dac	4fd191d6	b0530c50	fbad5c96
0	0	41	74	1	6160	183	6	17	37	0	2	0	17	68fd1e64	c1384774	2ed12532	6af132a7	25c83c98	7e0ccccf

模型代码地址：https://github.com/sql-machine-learning/elasticdl/tree/develop/model_zoo/dac_ctr

哈希变换

标准化和分箱

模型性能

CTR 算法	测试集 logloss
xDeepFM	0.45634
Wide & Deep	0.45998
Deep Cross Network	0.45988
Kaggle Best	0.44463

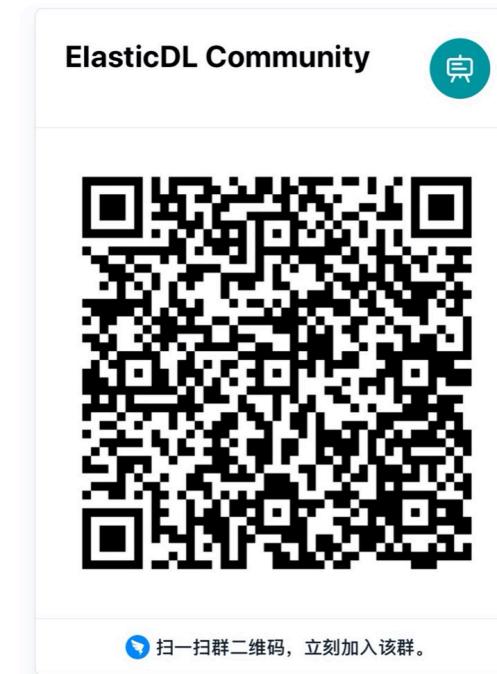


总结：

- ElasticDL 基于 TensorFlow API 提供了分布式深度学习编程框架，降低分布式程序编程难度。
- ElasticDL 基于 Kubernetes 和 TensorFlow 实现了弹性分布式训练，来提高作业研发效率和集群资源利用率。
- ElasticDL 针对蚂蚁集团 CTR 预估场景，提供了包括特征处理和模型训练的开箱即用方案，已在蚂蚁集团多个搜索推荐场景落地。

ElasticDL 项目 github 地址

<https://github.com/sql-machine-learning/elasticdl>



Q & A