
KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2019

Large Scale Distributed Deep Learning on Kubernetes Clusters

Yuan Tang – Ant Financial
Yong Tang – MobileIron

Speakers

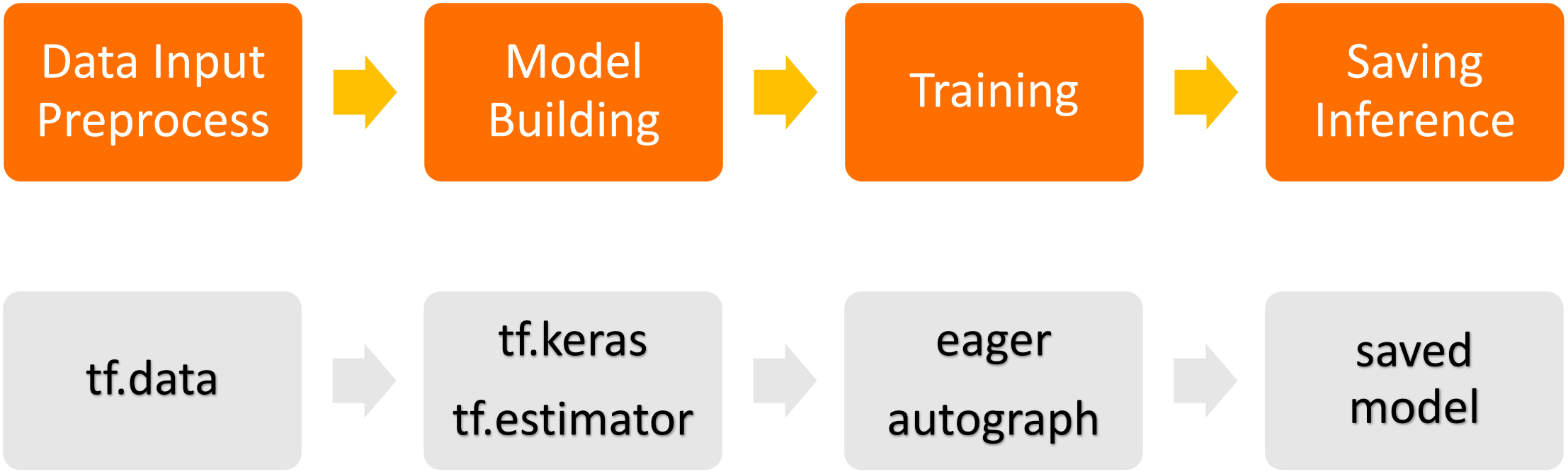
- Yuan Tang
  - GitHub: terrytangyuan
  - Member: Kubeflow
  - Maintainer: MXNet, XGBoost, TensorFlow
  - Senior Software Engineer, Ant Financial
- Yong Tang
  - GitHub: yongtang
  - Maintainer: CoreDNS and Docker
  - Director of Engineering, MobileIron

TensorFlow

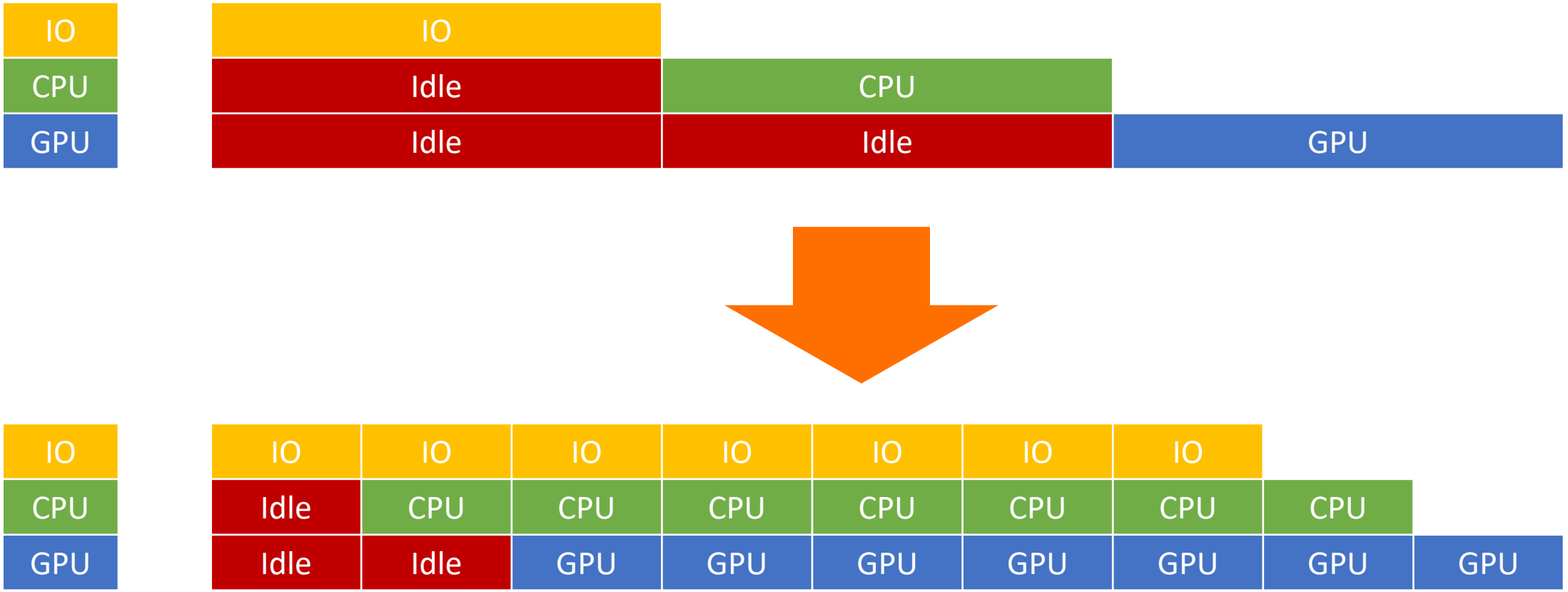
kubernetes

Kubeflow

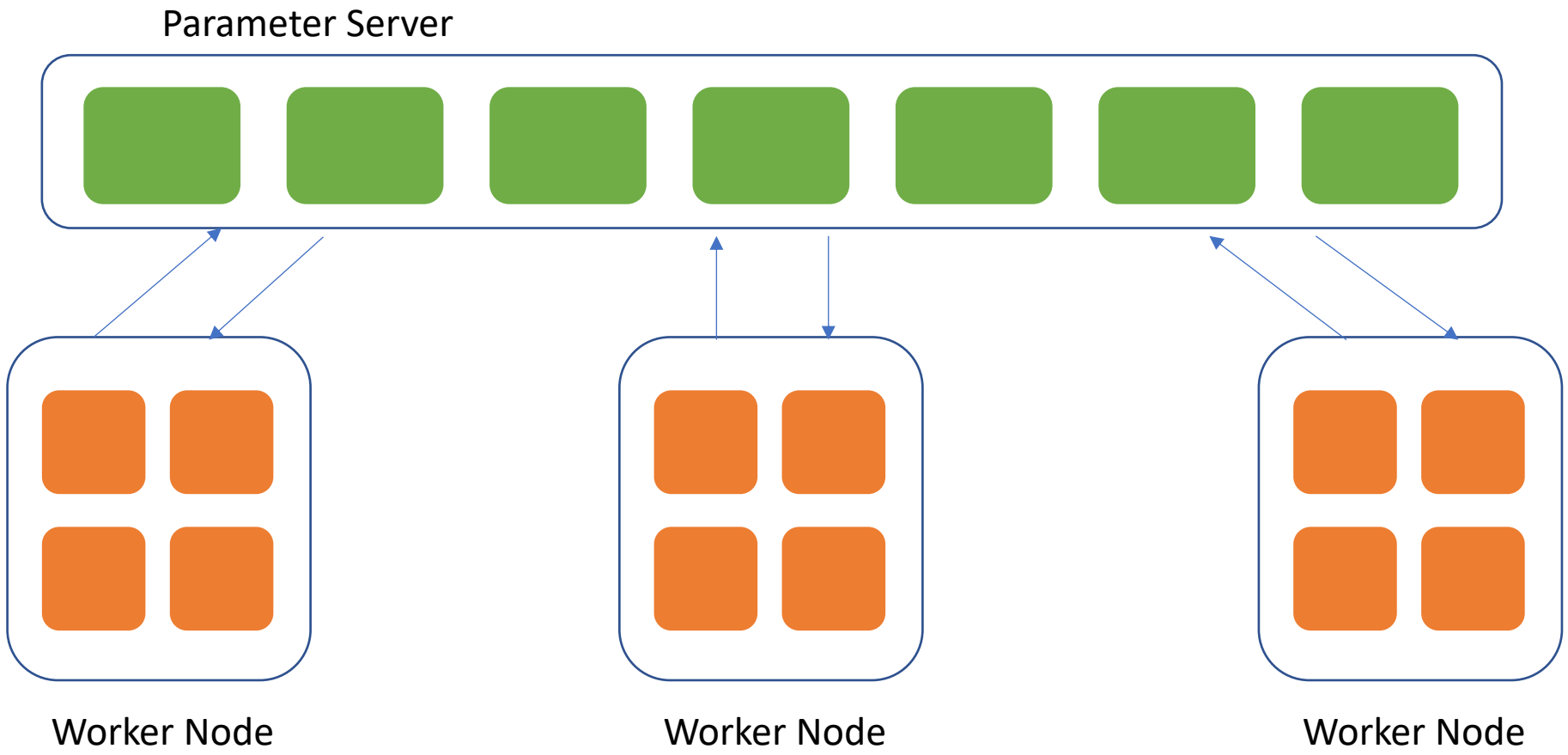
TensorFlow 2.0 Workflow



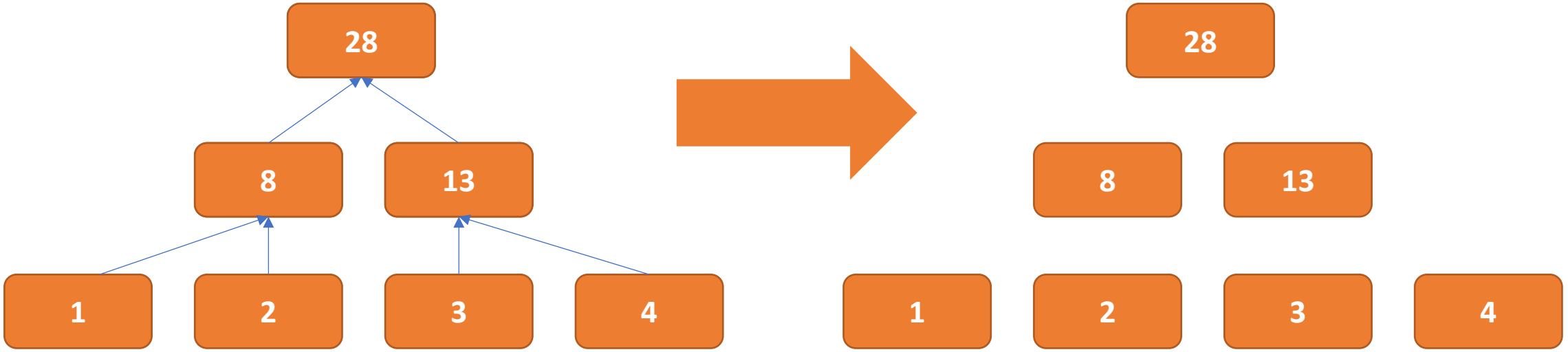
Orchestration for DL



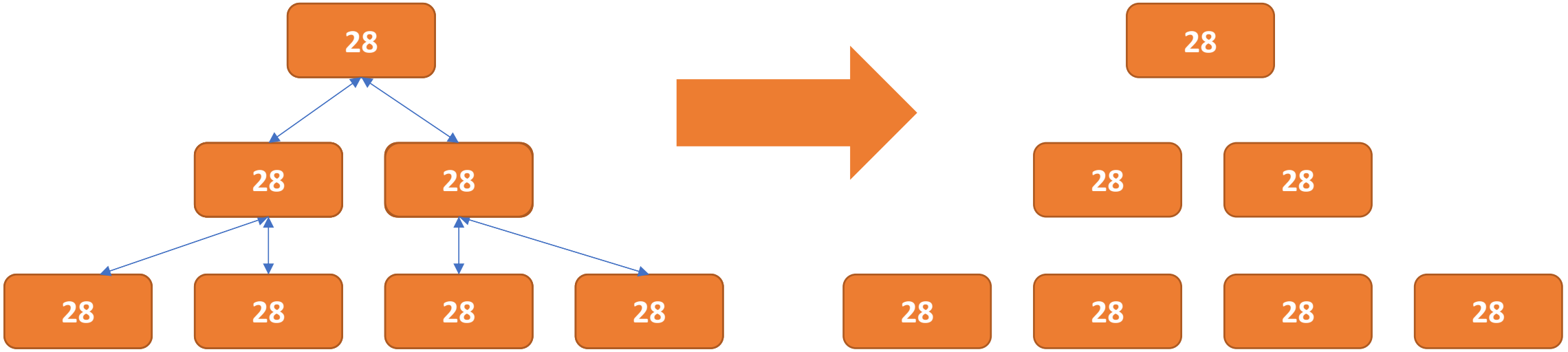
Parameter Server



Reduce



AllReduce



AllReduce == Reduce + Broadcast

Parameter Server

- Parallelize on a machine
- Parallelize in a cluster
- Controversial
- Cross device communication cost
- Huge efforts invested over the years

Orchestration for DL

Stateful Metadata

Lifecycle Management

Kubernetes for Orchestration

Kubernetes Operators for ML

Kubernetes Operators

TF Operator

PyTorch Operator

MPI Operator

Kubernetes Operators

	TF Operator	PyTorch Operator	MPI Operator
Framework Support	TensorFlow	PyTorch	HOROVOD TF/Keras/MXNet/PyTorch OpenMPI
Distribution Strategy & Backend	tf.distribute MPI/NCCL/PS/TPU	torch.distributed Gloo/MPI/NCCL	horovod DistributedOptimizer (MPI Only)

TFJob vs. MPIJob


```
apiVersion: "kubeflow.org/v1beta1”
kind: TFJob
metadata:
  name: distributed-training
spec:
 tfReplicaSpecs:
    Worker:
      replicas: 4
      template:
        spec:
          containers:
          - name: tensorflow
            image: distributed_training_tf:latest
            resources:
              limits: nvidia.com/gpu: 4
            command: "python tf_benchmarks.py"
```

```
apiVersion: "kubeflow.org/v1alpha2”
kind: MPIJob
metadata:
  name: distributed-training
spec:
 mpiReplicaSpecs:
    Worker:
      replicas: 4
      template:
        spec:
          containers:
          - name: tensorflow
            image: distributed_training_hovorod:latest
            resources:
              limits: nvidia.com/gpu: 4
            command: ”mpirun python hovorod_benchmarks.py"
```

TensorFlow


```
$ pip install tensorflow
$ pip install tensorflow-io

import tensorflow as tf
import tensorflow_io.mnist as mnist_io

dataset = mnist_io.MNISTDataset(image_filenames, label_filenames)
dataset = dataset.map(
   lambda x, y: (tf.image.convert_image_dtype(x, tf.float32), y)).batch(1000)

model = tf.keras.models.Sequential([
   tf.keras.layers.Flatten(input_shape=(28, 28)),
   tf.keras.layers.Dense(512, activation=tf.nn.relu),
   tf.keras.layers.Dropout(0.2),
   tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])
model.compile(loss='mse', optimizer='sgd’)
model.fit(dataset, epochs=2000)
model.evaluate(dataset)
```

Mirror Strategy in TF


```
import tensorflow as tf
import tensorflow_io.mnist as mnist_io

dataset = mnist_io.MNISTDataset(...)

model = tf.keras.Sequential([...])

mirrored_strategy = tf.distribute.MirroredStrategy()
with mirrored_strategy.scope():
   model.compile(loss='mse', optimizer='sgd’)

model.fit(dataset, epochs=2000)
model.evaluate(dataset)
```

TensorFlow + Hovorod


```
import tensorflow as tf
import tensorflow_io.mnist as mnist_io
import horovod.keras as hvd

dataset = mnist_io.MNISTDataset(...)

model = tf.keras.Sequential([...])

opt = tf.train.AdagradOptimizer(0.01 * hvd.size())
opt = hvd.DistributedOptimizer(opt)

model.compile(loss='mse', optimizer=opt)

callbacks = [
    hvd.callbacks.BroadcastGlobalVariablesCallback(0),
]
model.fit(dataset, epochs=2000, callbacks=callbacks)
model.evaluate(dataset)
```

TensorFlow vs. Horovod


```
import tensorflow as tf
import tensorflow_io.mnist as mnist_io

dataset = mnist_io.MNISTDataset(...)

model = tf.keras.Sequential([...])

mirrored_strategy = tf.distribute.MirroredStrategy()
with mirrored_strategy.scope():
   model.compile(loss='mse', optimizer='sgd’)

model.fit(dataset, epochs=2000)
model.evaluate(dataset)
```

```
import tensorflow as tf
import tensorflow_io.mnist as mnist_io
import horovod.keras as hvd

dataset = mnist_io.MNISTDataset(...)

model = tf.keras.Sequential([...])

opt = tf.train.AdagradOptimizer(0.01 * hvd.size())
opt = hvd.DistributedOptimizer(opt)

model.compile(loss='mse', optimizer=opt)

callbacks = [
   hvd.callbacks.BroadcastGlobalVariablesCallback(0),
]
model.fit(dataset, epochs=2000, callbacks=callbacks)
model.evaluate(dataset)
```

PyTorch + Hovorod


```
import torch
import horovod.torch as hvd

data_loader = torch.utils.data.DataLoader(train_dataset, batch_size=100)

model = ...

optimizer = torch.optim.SGD(model.parameters())
optimizer = hvd.DistributedOptimizer(
   optimizer, named_parameters=model.named_parameters())
hvd.broadcast_parameters(model.state_dict(), root_rank=0)

for epoch in range(100):
   for batch_idx, (data, target) in enumerate(data_loader):
      optimizer.zero_grad()
      output = model(data)
      loss = torch.nn.functional.F.nll_loss(output, target)
      loss.backward()
      optimizer.step()
```

Recall: TFJob vs. MPIJob


```
apiVersion: "kubeflow.org/v1beta1”
kind: TFJob
metadata:
  name: distributed-training
spec:
 tfReplicaSpecs:
    Worker:
      replicas: 4
      template:
        spec:
          containers:
          - name: tensorflow
            image: distributed_training_tf:latest
            resources:
              limits: nvidia.com/gpu: 4
            command: "python tf_benchmarks.py"
```

```
apiVersion: "kubeflow.org/v1alpha2”
kind: MPIJob
metadata:
  name: distributed-training
spec:
 mpiReplicaSpecs:
    Worker:
      replicas: 4
      template:
        spec:
          containers:
          - name: tensorflow
            image: distributed_training_hovorod:latest
            resources:
              limits: nvidia.com/gpu: 4
            command: ”mpirun python hovorod_benchmarks.py"
```

Shared API and Best Practices

KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2019

kubeflow / common

Unwatch 9 | Unstar 11 | Fork 8

Code | Issues 6 | Pull requests 2 | Projects 0 | Wiki | Insights

Common APIs and libraries shared by other Kubeflow operator repositories.

39 commits | 1 branch | 0 releases | 5 contributors | Apache-2.0

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

terrytangyuan and k8s-ci-robot Correct function names in the comment (#32) — Latest commit f38f5dc 2 days ago

client	Common job controller library (#5)	28 days ago
hack	Common job controller library (#5)	28 days ago
job_controller	Correct function names in the comment (#32)	2 days ago
operator/v1	Fix incorrect name for restart policy exit code (#20)	10 days ago
test_job/v1	Remove mentions of tensorflow in test job (#21)	10 days ago
test_util/v1	chore: Fix package name (#27)	8 days ago
util	Move public util functions to util/status.go	6 days ago
.gitignore	Added .gitignore file (#16)	15 days ago
.travis.yml	Update goveralls ignore pattern	9 days ago
LICENSE	Create LICENSE	a month ago
OWNERS	Add terrytangyuan to OWNERS	a month ago
README.md	Add Travis badge and Go report card (#9)	27 days ago

Shared API and Best Practices

- Common and standardized API spec
- Base JobController interface
- JobController implementation utilities
- Testing utilities

THANK YOU