WenQuanYi Micro Hei [Scale=0.9]WenQuanYi Micro Hei Mono song-WenQuanYi Micro Hei sfWenQuanYi Micro Hei "zh" = 0pt plus 1pt

FleetX

åŔŚåÿČ 0.1.0.beta

PaddlePaddle

2020 åźť 11 ælJĹ 24 æŮě

å£ńéĂ§åijĂåğŃ

- æňćè£ŐæĆĺåĚşæşĺéčđæąĺåĹĘåÿČåijŕèő■çżČïijŇæĹŚäžňåÿŇæIJŻèČ¡åÿőåŁľæŕŔäÿĂäÿłçŤĺæĹůèţřäÿŁ

CHAPTER 1

æŢťä¡ŞäżŃçż■äÿŐåĘĚåőźæęĆèğĹ

1.1 æňćè£ŐåĚşæşĺåď˙ğèğĎæĺąæůśåžęå■ęäźăæŁĂæIJŕ

è£Śå■Ąåźť˙æİěïijŇæůśåžęå■ęäźăæŁĂæIJŕäÿ■æŰ■åĹůæŰřèğĘèğL˙ãĂĄèĞłçĎűèŕ■èĺĂãĂĄèŕ■é§şãĂĄæŘ

1.2 éčđæąĺåĹĘåÿČåijŔèő■çżČæŔŘä¿ŻçŽĎæăÿå£ČäžůåĂij

1. æžŘèĞłäžğäÿŽåőđèůţçŽĎçżŔélŇïijŽ
   - éčđæąĺçŽĎåĹĘåÿČåijŔèő■çżČæŁĂæIJŕæžŘèĞłçŹ¿åžęçŽĎäÿŽåŁąåőđèůţïijŇæŸŕçżŔè£ĞèűĚåď˙ğèğĎ
   - éčđæąĺåĹĘåÿČåijŔèő■çżČçżŔè£Ğåőđèůţæč˘ĂélŇçŽĎåžŤçŤĺéćĘå§§åŇĚæŃňèĞłçĎűèŕ■èĺĂåď˙ĎçŘĘïij
2. åőŇåď˙ĞçŽĎåźűèąŇæĺąåijŔïijŽ
   - æŢřæ■őåźűèąŇïijŽéŠĹåŕźäžğäÿŽçŢŇæIJĂåÿÿçŤĺçŽĎæŢřæ■őåźűèąŇæĺąåijŔïijŇéčđæąĺéŠĹåŕźåőđéŹĚ
   - æţĄæřť˙çž£åźűèąŇïijŽéİćåŘŚåijĆæđĎçąňäžűïijŇæţĄæřť˙çž£åźűèąŇèČ¡åď˙§åřĘæĺąåđŃèőąçőŮéČĺåĹĘą
   - æĺąåđŃåźűèąŇïijŽåŕźäžŐèűĚåď˙ğèğĎæĺąåĹĘçśžéŮőécŸïijŇéčđæąĺæŔŘä¿Żèőąçő̊ŮäÿŐå■ŸåĆĺåŘŇæŮ
3. éİćåŘŚäžŚçńŕåIJžæŹŕçŽĎåźűèąŇèő■çżČçżĎäžűïijŽ
   - éčđæąĺéŠĹåŕźéŻĘç¿d˙ç¡ŚçżIJçŐŕåćČãĂĄçąňäžűèő¿åď˙ĞæŕŤè¿Čä¡ŐéĚ■çŽĎåIJžæŹŕæŔŘä¿Żåď˙Žçğ■å
   - éŠĹåŕźäžŚçńŕçőŮåŁŻåĚůæIJL˙åijźæĂğçŽĎçL˙źçĆźïijŇéčđæąĺäź§åğŃçżĹåIJĺæŐćçť˙ćåijźæĂğæůśåžęå■

1.3 åijĂåğŃä¡ăçŽĎåĹĘåÿČåijŔèő■çżČäźŃæŮĚ

- æŢťä¡ŞåĘĚåőźïijŽæĹŚäžňæŐĺè■ŘæĆĺçŻťæŐěæăźæ■őäÿżéąţïijŇæŇL'çĚğçńäèŁĆéąžåžŔéĂŘäÿłæţŔè
- FAQïijŽåŕźäžŐéńŸéćŚåĞžçŐřçŽĎéŮőéćŸïijŇæĹŚäžňäijŽåőŽæIJ§æŢť'çŘEçŻÿåĚşåĘĚåőźåĹřFAQ
- å£ńéĂ§äÿŁæL'ŃïijŽåęĆæđIJæČşæIJĂä¡ŐæĹŘæIJňçŽĎäžĘèğčéčđæąĺçŽĎåĹĘåÿČåijŔèő■çżČïijŇæĹŚ
- GPUåď'ŽæIJžèő■çżČïijŽåęĆæđIJæĆĺåůšçżŔåijĂåğŃä¡£çŤĺGPUè£ŻèąŇåď'ŽæIJžåď'Žå■ąèő■çżČïijŇæ
- åŔĆæŢřæIJ■åŁąåŹĺïijŽä£ąæĄŕæčĂçť'ćãĂĄæŐĺè■ŘçşżçżšécĘå§§åÿÿçŤĺçŽĎåźűèąŇèő■çżČæŰźåijŔïi
- åĚňæIJL'äžŚçŐŕåćČåőđèůţïijŽåęĆæđIJæĆĺåIJĺåĚňæIJL'äžŚäÿŁèůŚèĞłåůšçŽĎGPUåď'Žå■ąäżżåŁąïijŇa
- åijźæĂğèő■çżČïijŽåęĆæđIJåŕźåęĆä¡ŢåĹľçŤĺäžŚçńŕåijźæĂğèţĎæžŘè£ŻèąŇåď'ğèğĎæĺąèŠÿéęŔèő■çżČ
  æIJ■åŁąåđŃåijźæĂğèŠÿéęŔ

1.4 RoadMap

- æĹŚäžňäź§äijŽæŐĺéĂĄåď'ğèğĎæĺąæůšåžęå■ęäźăæŁĂæIJŕécĘå§§æIJĂåL'■æš£çŽĎæŁĂæIJŕåĹřè£ŹéČ
  - è£ŚæIJ§ïijŽå■Čäž£èğĎæĺąæĺąåđŃåŘĆæŢřçŽĎGPUåď'ŽæIJžåď'Žå■ąèő■çżČïijŇæŢňèřůæIJ§å¿Ě

CHAPTER 2

åőL'èčĚPaddleäÿŐFleetX

2.1 Paddle

ä¡£çŤĺéčđæąĺè£ŻèąŇåĹĘåÿČåijŔèő■çżČçŽĎæIJĂářŔåőL'èčĚéŻĘåŘĹářśæŸŕåőL'èčĚPaddleãĂĆäżŐPad
2.0çL'ĹæIJňåijĂåğŃïijŇæĹŚäżňéİćåŘŚäÿ■åŘŇçŤĺæĹůç¿d'ä¡ŞæŔŘä¿Żäÿ■åŘŇçśżåđŃçŽĎåĹĘåÿČåijŔèő■ç

- éİćåŘŚçőŮæşŢåůěçĺŃåÿĹäÿžäÿżçŽĎéńŸçžğAPI **paddle.distributed.fleet**ãĂĆ
- éİćåŘŚåĚůæIJL'åĹĘåÿČåijŔèő■çżČåžŢåśĆåůěçĺŃåijĂåŘŚèČ¡åŁŻçŽĎåůěçĺŃåÿĹæŔŘä¿ŻçŽĎAPI **paddle.distributed**ãĂĆ
- æĆĺåŔłéIJĂèęĄåőL'èčĚPaddleïijŇårśåŔŕäżèèŐůå¿ŮéčđæąĺåŻćéŸ§åőŸæŰźæŔŘä¿ŻçŽĎæL'ĂæIJL'åĹ

```
pip install paddlepaddle-gpu
```

åĚşäžŐåőL'èčĚPaddleïijŇè£ŹéĞŇ æIJL'æŻť'åőŇåď'ĞçŽĎåőL'èčĚæŇĞå■Ůä¿ŻæĆĺåŔĆèĂČãĂĆ

2.2 FleetX

æŻť'åď'ğèğĎæĺąçŽĎæŢřæ■őãĂĄèČ¡åď'§èőřå£ĘåźűæşŻåŇŰåď'ğæŢřæ■őçŽĎæĺąåđŃãĂĄèűĚåď'ğèğĎæ
FleetX åůěåĚůåŇĚãĂĆ

- åIJĺæŢřæ■őçżť'åžęïijŇæŔŘä¿ŻçŤĺæĹůåŔŕäżěå£ńéĂ§åőŽäźL'çŽĎæăĞåĞĘåĚňåijĂæŢřæ■őéŻĘäżěåŔŁ
- åIJĺæĺąåđŃçżť'åžęïijŇFleetXæŔŘä¿ŻåĚÿåđŃçŽĎåĹĘåÿČåijŔèő■çżČåIJžæŹŕäÿŃæIJĂåÿÿçŤĺçŽĎæăĞ
- åIJĺåĹl'çŤĺåď'ğèğĎæĺąçőŮåŁŻéŻĘç¿d'æŰźéİćïijŇFleetXä¡£çŤĺPaddleåŐ§çŤ§æŔŘä¿ŻçŽĎåĹĘåÿČåijĺ

```
pip install fleet-x==0.0.7
```

æĹŰèĂĚä¡£çŤĺæĹŚäžňäÿžçŤĺæĹůæŔŘä¿ŻäžĘåůšçżŔçijŰèŕŚåě¡çŽĎåőL'èčĚåŇĚïijŇåŔŕäžěäÿŃè¡¡åĹřa

```
# python2
wget --no-check-certificate https://fleet.bj.bcebos.com/fleet_x-0.0.
↪7-py2-none-any.whl
pip install fleet_x-0.0.4-py2-none-any.whl
# python3
wget --no-check-certificate https://fleet.bj.bcebos.com/fleet_x-0.0.
↪7-py3-none-any.whl
pip3 install fleet_x-0.0.4-py3-none-any.whl
```

CHAPTER 3

éİŹæĂĄåŻ¿åĹĘåÿČåijŔèő■çżČå£ńéĂ§åijÅågŃ

åŕźäžŐåď ğéČĺåĹEçŤĺæĹůæİěèőšïijŇæŢřæ■őåźűèąŇèő■çżČå§žæIJňåŔŕäžěèğčåĘşåőđéŹĚäÿŽåŁąäÿ■çŽ
paddle.distributed.fleetè£ŻèąŇæŢřæ■őåźűèąŇèő■çżČãĂĆåIJĺæŢřæ■őåźűèąŇæŰźåijŔäÿŃïijŇéA
TrainingïijL’åŠŇåŔĆæŢřæIJ■åŁąåŹĺèő■çżČïijĹParameter Server Train-
ingïijL’ïijŇæŐěäÿŃæİěçŽĎä¿Ńå■ŘäijŽäžěåŘŇæăůçŽĎæĺąåđŃæİěèŕť æŸŐäÿď çğ■æđűæđĎçŽĎæŢřæ■őåźűè

3.1 çL’ĹælJňèęĄæśĆ

- paddlepaddle-2.0.0-rc-cpu / paddlepaddle-2.0.0-rc-gpuåŘŁäžěäÿŁ

3.2 æĺąåđŃæŘŔè£ř

äÿžäžĘæŰźä¿£èŕť æŸŐïijŇæĹŚäžňéĞĞçŤĺäÿď åśĆåĚĺè£đæŐěç¡ŚçżIJçŽĎåĹEçśżæĺąåđŃïijŇåźűä¡£çŤĺ
py

```
import paddle
import paddle.static.nn as nn

paddle.enable_static()
def mnist_on_mlp_model():
    train_dataset = paddle.vision.datasets.MNIST(mode='train')
    test_dataset = paddle.vision.datasets.MNIST(mode='test')
    x = paddle.data(name="x", shape=[64, 1, 28, 28], dtype='float32
↪')
    y = paddle.data(name="y", shape=[64, 1], dtype='int64')
    x_flatten = paddle.reshape(x, [64, 784])
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
    fc_1 = nn.fc(input=x_flatten, size=128, act='tanh')
    fc_2 = nn.fc(input=fc_1, size=128, act='tanh')
    prediction = nn.fc(input=[fc_2], size=10, act='softmax')
    cost = paddle.fluid.layers.cross_entropy(input=prediction,␣
↪label=y)
    acc_top1 = paddle.fluid.layers.accuracy(input=prediction,␣
↪label=y, k=1)
    avg_cost = paddle.fluid.layers.mean(x=cost)
    return train_dataset, test_dataset, x, y, avg_cost, acc_top1
```

3.3 éĞĞçŤÍGPUåďŽælJžåďŽå■ąè£ŻèąŇåŘŇæ■ěèő■çżČ

collective_trainer.py

```
import os
import paddle
import paddle.distributed.fleet as fleet
from model import mnist_on_mlp_model

train_data, test_data, x, y, cost, acc = mnist_on_mlp_model()
place = paddle.CUDAPlace(int(os.environ.get('FLAGS_selected_gpus',␣
↪0)))
train_dataloader = paddle.io.DataLoader(
    train_data, feed_list=[x, y], drop_last=True,
    places=place, batch_size=64, shuffle=True)
fleet.init(is_collective=True)
strategy = fleet.DistributedStrategy()
#optimizer = paddle.optimizer.Adam(learning_rate=0.01)
optimizer = paddle.fluid.optimizer.Adam(learning_rate=0.001)
optimizer = fleet.distributed_optimizer(optimizer,␣
↪strategy=strategy)
optimizer.minimize(cost)

exe = paddle.static.Executor(place)
exe.run(paddle.static.default_startup_program())

epoch = 10
step = 0
for i in range(epoch):
    for data in train_dataloader():
        step += 1
        loss_val, acc_val = exe.run(
          paddle.static.default_main_program(),
          feed=data, fetch_list=[cost.name, acc.name])
```

- å■ŢæIJžåŻŽå■ąèő■çżČåŘŕåŁĺåŚ¡äžď

```
fleetrun --gpus 0,1,2,3 collective_trainer.py
```

3.4 éĞĞçŤĺåŔĆæŢřælJ■åŁąåŹÍè£ŻèąŇåďŽælJžèő■çżČ

parameter_server_trainer.py

```
import paddle
import paddle.distributed.fleet as fleet
from model import mnist_on_mlp_model

paddle.enable_static()

train_data, test_data, x, y, cost, acc = mnist_on_mlp_model()

fleet.init()
strategy = fleet.DistributedStrategy()
strategy.a_sync = True
optimizer = paddle.fluid.optimizer.Adam(learning_rate=0.001)
optimizer = fleet.distributed_optimizer(optimizer, strategy)
optimizer.minimize(cost)

if fleet.is_server():
   fleet.init_server()
   fleet.run_server()
else:
   place = paddle.CPUPlace()
   exe = paddle.static.Executor(place)
   exe.run(paddle.static.default_startup_program())
   fleet.init_worker()

   train_dataloader = paddle.io.DataLoader(
      train_data, feed_list=[x, y], drop_last=True, places=place,
      batch_size=64, shuffle=True)

   epoch = 1
   for i in range(epoch):
      for data in train_dataloader():
         cost_val, acc_val = exe.run(
            paddle.static.default_main_program(),
            feed=data, fetch_list=[cost.name, acc.name])
         print("loss: {}, acc: {}".format(cost_val, acc_val))
   fleet.stop_worker()
```

- äÿď'èŁĆçĆźServerïijÑäÿď'èŁĆçĆźWorkerçŽĎåŘŕåŁĺåŚ¡äżď'

```
fleetrun --worker_num 2 --server_num 2 parameter_server_trainer.py
```

CHAPTER 4

åŁÍæĂĄåŻ¿åĹĘåÿČåijŔèő■çżČå£ńéĂ§åijĂåğŃ

PaddleåőŸæŰźæŰĞæąčäÿ■åŕźåŁÍæĂĄåŻ¿ïijĹåŚ¡äžd'åijŔçijŰçÍŃïijL'åĄŽäžĘæŕŤè¿Čèŕęçżę̧ŽĎäżŃçż
paddle.distributed.fleetæŐěåŔčäžŐPaddle 2.0-RCçL'ÍæIJňåijĂåğŃæŤŕæŇĄåŁÍæĂĄåŻ¿åĹĘåÿČ
paddle.distributed.fleetæŐěåŔčè£ŻèąŇåŁÍæĂĄåŻ¿åĹĘåÿČåijŔèő■çżČãĆĆæŐěäÿŃæİěæĹŚäžň

æşÍïijŽçŻőåL'■paddle.distributed.fleetåŘŕåŁĺåŁÍæĂĄåŻ¿åĹĘåÿČåijŔèő■çżČäžĚæŤŕæŇĄé
CommunicationïijL'æÍąåijŔïijŇäÿ■æŤŕæŇĄåŔĆæŢŕæIJ■åŁąåŹĺïijĹParameter-
ServerïijL'æÍąåijŔãĂĆæIJňæŰĞçd'žä¿ŃäÿžéŻĘåŘĹéĂŽä£ąæÍąåijŔäżżåŁąãĂĆ

4.1 çL'ÍæIJňèęĄæśĆ

- paddlepaddle 2.0-rc-gpuçL'ÍæIJňåŔŁäžěäÿŁ

4.2 å■ŢæIJžå■Ţå■ąèő■çżČ

äÿŃéİćæŸŕäÿĂäÿłéİđåÿÿçőĂå■ŢçŽĎåŁÍæĂĄåŻ¿å■ŢæIJžå■Ţå■ąçÍŃåžŔãĂĆç¡ŚçżIJåŔłæIJL'åŔłæIJL'2

```
# -*- coding: UTF-8 -*-
import paddle
import paddle.nn as nn

# åőŽäźL'åĚĺè£đæŐěç¡ŚçżIJïijŇéIJĂçżğæL'£èĞłnn.Layer
class LinearNet(nn.Layer):
    def __init__(self):
        super(LinearNet, self).__init__()
        self._linear1 = nn.Linear(10, 10)
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
        self._linear2 = nn.Linear(10, 1)

    def forward(self, x):
        return self._linear2(self._linear1(x))


# 1.åijĂåŘŕåŁĺæĂĄåŻ¿æĺąåijŔ
paddle.disable_static()

# 2. åőŽäźL'ç¡ŚçżIJåŕźèśąïijŇæ■$åd'śåĞ¡æŢřåŠŇäijŸåŇŰåŹĺ
layer = LinearNet()
loss_fn = nn.MSELoss()
adam = paddle.optimizer.Adam(
learning_rate=0.001, parameters=layer.parameters())


for step in range(20):
    # 3. æL'ğèąŇåL'■åŘŚç¡ŚçżIJ
    inputs = paddle.randn([10, 10], 'float32')
    outputs = layer(inputs)
    labels = paddle.randn([10, 1], 'float32')
    loss = loss_fn(outputs, labels)

    print("step:{}\tloss:{}".format(step, loss.numpy()))

    # 4. æL'ğèąŇåŔ■åŘŚèőąçőŮåŠŇåŔĆæŢřæŻt'æŰř
    loss.backward()
    adam.step()
    adam.clear_grad()
```

åřĘäžěäÿŁäžčçăĄä£Ĺå■Ÿäÿžtrain_single.pyïijŇè£ŘèąŇpython train_single.pyïijŇæĆĺåřĘçIJŃåĹřæŸ¿çd'žåęĆäÿŃæŮěå£Ůä£ąæĄŕïijŽ

```
step:0  loss:[1.2709768]
step:1  loss:[0.7705929]
step:2  loss:[2.2044802]
step:3  loss:[1.6021421]
step:4  loss:[2.0286825]
step:5  loss:[0.7866151]
step:6  loss:[1.926115]
step:7  loss:[0.3647427]
...
```

4.3 å■ŢælJžåď'Žå■ąèő■çżČ

ä¡£çŤĺFleetæŐěåŔčè£ŻèąŇåŁĺæĂĄåŻ¿åĹĘåÿČåijŔèő■çżČåĚűåőđéĺđåÿÿçŽĎçőĂå■ŢïijŇåŔłéIJĂä£őæ

1. åŕijåĚěpaddle.distributed.fleetåŇĚ

```
from paddle.distributed import fleet
```

2. åĹİåğŃåŇŰfleetçŐŕåćČ

```
fleet.init(is_collective=True)
```

3. éĂŽè£ĞfleetèŐůåŔŰåĹĘåÿČåijŔäijŸåŇŰåŹĺåŠŇåĹĘåÿČåijŔæĺąåđŃ

```
strategy = fleet.DistributedStrategy()
adam = fleet.distributed_optimizer(adam, strategy=strategy)
dp_layer = fleet.distributed_model(layer)
```

èŕť'æŸŐïijŽçŻőåL'■éİŹæĂĄåŻ¿DistributedStrategyäÿŃçŽĎåĹĘåÿČåijŔç■ŰçŢěæ■čéĂŘæ■ěåŀ

æăźæ■őæĹŚäžňæIJĂåijĂåğŃæŔŘä¿ŻçŽĎå■ŢæIJžå■Ţå■ąäžčçăĄçđ'žä¿ŃïijŇåĘ■æăźæ■ő3æ■ěåŔčèŕĂè‡

```
# -*- coding: UTF-8 -*-
import paddle
import paddle.nn as nn
#åĹĘåÿČåijŔstep 1: åŕijåĚěpaddle.distributed.fleetåŇĚ
from paddle.distributed import fleet

# åőŽäźL'åĚĺè£đæŐěç¡ŚçżIJïijŇéIJĂçżğæL'£èĞłnn.Layer
class LinearNet(nn.Layer):
    def __init__(self):
        super(LinearNet, self).__init__()
        self._linear1 = nn.Linear(10, 10)
        self._linear2 = nn.Linear(10, 1)

    def forward(self, x):
        return self._linear2(self._linear1(x))


# 1.åijĂåŘŕåŁĺæĂĄåŻ¿æĺąåijŔ
paddle.disable_static()

# åĹĘåÿČåijŔstep 2: åĹİåğŃåŇŰfleet
fleet.init(is_collective=True)

# 2. åőŽäźL'ç¡ŚçżIJåŕźèśąïijŇæ■Šåd'śåĞ¡æŢŕåŠŇäijŸåŇŰåŹĺ
layer = LinearNet()
loss_fn = nn.MSELoss()
adam = paddle.optimizer.Adam(
    learning_rate=0.001, parameters=layer.parameters())

# åĹĘåÿČåijŔstep 3:␣
↪éĂŽè£ĞfleetèŐůåŔŰåĹĘåÿČåijŔäijŸåŇŰåŹĺåŠŇåĹĘåÿČåijŔæĺąåđŃ
strategy = fleet.DistributedStrategy()
adam = fleet.distributed_optimizer(adam, strategy=strategy)
dp_layer = fleet.distributed_model(layer)
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
for step in range(20):
    # 3. æL'ğèąŇåL'■åŘŠç¡ŚçżIJ
    inputs = paddle.randn([10, 10], 'float32')
    outputs = dp_layer(inputs)
    labels = paddle.randn([10, 1], 'float32')
    loss = loss_fn(outputs, labels)

    print("step:{}\tloss:{}".format(step, loss.numpy()))

    # 4. æL'ğèąŇåŕ■åŘŚèőąçőŮåŠŇåŕĆæŢřæŻt'æŰř
    loss.backward()

    adam.step()
    adam.clear_grad()
```

åřĘäžěäÿŁäžčçăĄä£Ìå■Ÿäÿžtrain_fleet.pyïijŇåĄĞèő¿èęĄè£ŘèąŇ2å■ąçŽĎäżżåŁąïijŇéĆčäźĹåŕ

```
fleetrun --gpus=0,1 dygraph_fleet.py
```

æĆĺåřĘçIJŃåĹřæŸ¿çd'žåęĆäÿŃæŮěå£Ůä£ąæĄŕïijŽ

```
----------  Configuration Arguments ----------
gpus: 0,1
ips: 127.0.0.1
log_dir: log
server_num: None
servers:
training_script: dygraph_fleet.py
training_script_args: []
worker_num: None
workers:
------------------------------------------------
INFO 2020-0X-XX 08:33:30,247 launch.py:441] Run collective gpu mode.
 ↪ gpu arguments:['--gpus'], cuda count:8
INFO 2020-0X-XX 08:33:30,247 launch_utils.py:430] Local start 2␣
 ↪processes. First process distributed environment info (Only For␣
 ↪Debug):
   ␣
 ↪+=======================================================================================
    |                        Distributed Envs                              ␣
 ↪Value                      |
    +-----------------------------------------------------------------
 ↪----------------------+
    |                  PADDLE_CURRENT_ENDPOINT                  127.0.
 ↪0.1:59664               |
    |                      PADDLE_TRAINERS_NUM                          ␣
 ↪2                       |
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
   |                       FLAGS_selected_gpus                              ␣
↪0                        |
   |                   PADDLE_TRAINER_ENDPOINTS         127.0.0.
↪1:59664,127.0.0.1:48993       |
   |                         PADDLE_TRAINER_ID                             ␣
↪0                        |

 ␣
↪+=============================================================================
step:0  loss:[1.3279431]
step:1  loss:[2.5023699]
step:2  loss:[3.3197324]
step:3  loss:[2.6869867]
step:4  loss:[2.6306524]
step:5  loss:[1.9267073]
step:6  loss:[1.2037501]
step:7  loss:[1.1434236]
...
```

åőŇæŢť'2å■ąçŽĎæŮěå£Ůä£ąæĄŕäź§åŕŕåIJĺ . /log/çŻőå¡ŢäÿŃæ§ěçIJŃãĂĆäžĘèğčæŻť'åď'Žfleetru
åŘŕåŁĺåĹĘåÿČåijŔäżżåŁąãĂĆ

4.4 åď'ŽælJžåď'Žå■ąèő■çżČ

äżŐå■ŢæIJžåď'Žå■ąåĹřåď'ŽæIJžåď'Žå■ąèő■çżČïijŇåIJĺäžčçăĄäÿŁåźűäÿ■éIJĂèęĄåĄŽäżżä¡ŢæŤźåŁĺïijŇ

```
fleetrun --ips="xx.xx.xx.xx,yy.yy.yy.yy" --gpus=0,1 dygraph_fleet.py
```

åIJĺ2åŕŕæIJžåŹĺäÿŁ**åĹĘåĹń**è£ŘèąŇäżěäÿŁåŘŕåŁĺåŚ¡äżď'ïijŇfleetrunåřĘåIJĺåŘŐåŕŕåĹĘåĹńåŘŕåŁĺ
æĆĺåřĘåIJĺipäÿžxx.xx.xx.xxçŽĎæIJžåŹĺäÿŁçIJŃåĹřåŚ¡äżď'åŕŕè¿ŞåĞžæŮěå£Ůä£ąæĄŕïijŽ

```
----------  Configuration Arguments ----------
gpus: None
ips: xx.xx.xx.xx,yy.yy.yy.yy
log_dir: log
server_num: None
servers:
training_script: dygraph_fleet.py
training_script_args: []
worker_num: None
workers:
------------------------------------------------
INFO 2020-0X-XX 21:29:41,918 launch.py:434] Run collective gpu mode.
 ↪ gpu arguments:['--ips'], cuda count:2
INFO 2020-0X-XX 21:29:41,919 launch_utils.py:426] Local start 2␣
↪processes. First process distributed environment info (Only For␣
↪Debug):

 ␣
↪+=============================================================================
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
    |                          Distributed Envs                              ␣
↪Value                       |
    +-----------------------------------------------------------------------
↪----------------------+
    |                    PADDLE_CURRENT_ENDPOINT                 xx.xx.
↪xx.xx:6070              |
    |                        PADDLE_TRAINERS_NUM                           ␣
↪ 4                      |
    |                        FLAGS_selected_gpus                           ␣
↪ 0                      |
    |                   PADDLE_TRAINER_ENDPOINTS  ... :6071,yy.yy.yy.
↪yy:6070,yy.yy.yy.yy:6071|
    |                          PADDLE_TRAINER_ID                           ␣
↪ 0                      |
   ␣
↪+=======================================================================
step:0  loss:[5.2519045]
step:1  loss:[3.139771]
step:2  loss:[2.0075738]
step:3  loss:[1.4674551]
step:4  loss:[4.0751777]
step:5  loss:[2.6568782]
step:6  loss:[1.1998112]
...
```

åŘŇæăůåőŇæŢťçŽĎæŮěå£Ůä£ąæĄŕäź§åĹĘåĹńåIJĺxx.xx.xx.xxæIJžåŹĺåŠŇyy.yy.yy.yyæIJžåŹĺäÿŁçŽĎ
/log/çŻőå¡ŢäÿŃæ§ěçIJŃãĂĆ

4.5 åřŔçżŞ

èĞşæ■d'ïijŇçŻÿä£ąæĆĺåůšçżŔéĂŽè£Ğ3æ■ěåŔčèŕĂæŐŇæŔąäžĘåęĆä¡ŢåřĘäÿĂäÿłæŹőéĂŽçŽĎpaddle

CHAPTER 5

ä¡£çŤÍfleetrunåŘŕåŁÍåĹĘåÿČåijŔäżżåŁą

PaddleæŔŘä¿ŻåŚ¡äżd'èąŇåŘŕåŁÍåŚ¡äżd'`fleetrun`ïijŇéĚ■åŘĹPaddleçŽĎåĹĘåÿČåijŔénŸçžğAPI`pac`
`distributed.fleet`å■şåŔŕè¡żæİ¿åŘŕåŁÍPaddleéŻĘåŘĹéĂŽä£ąæÍąåijŔæĹŰåŔĆæŢřæIJ■åŁąåŹÍæÍąåijŔ
`fleetrun`åIJÍéİŹæĂĄåŻ¿åŠŇåŁÍæĂĄåŻ¿åIJžæŹŕäÿŃåİĞåŔŕä¡£çŤÍãĂĆ

5.1 åĘĚåőźåŕijèĹł

5.2 ä¡£çŤÍèęĄæśĆ

ä¡£çŤÍ`fleetrun`åŚ¡äżd'çŽĎèęĄæśĆ:

- åőL'èčĚ paddlepaddle 2.0-rc åŘŁäžěäÿŁ

5.3 ä¡£çŤÍèŕť'æŸŐ

`fleetrun`ä¡£çŤÍåIJžæŹŕäÿżèęĄåĹĘäÿžéŻĘåŘĹéĂŽä£ąèő■çżČïijĹCollective TrainingïijL'åŠŇåŔĆæŢřæIJ■åŁąåŹÍèő■çżČïijĹParameter Server TrainingïijL'ãĂĆ

éŻĘåŘĹéĂŽä£ąèő■çżČäÿĂèĹňåIJĺGPUèő¿åď ĞäÿŁè£ŘèąŇïijŇåŻäæ■ď æĹŠäžňåřĘäžŃçż■GPUå■ŢæIJžå■Ţ
åŘĆæŢřæIJ■åŁąåŹĺèő■çżČåŇĚåŘńæIJ■åŁąèŁĆçĆźãĂĄèő■çżČèŁĆçĆźäžěåŔŁåijĆæđĎèő■çżČèŁĆçĆźçŽĎ
åŽäæ■ď æĹŠäžňåřĘäžŃçż■åIJĺCPUéŻĘç¿d'ãĂĄGPUéŻĘç¿d'äÿŁåŠŇåijĆæđĎéŻĘç¿d'äÿŁåęĆä¡Ţä¡£çŤĺfle
ãĂĆfleetrunæŤŕæŇĄåIJĺçŹ¿åžęåĚňåŔÿåĘĚéČĺäžŚPaddleCloudäÿŁè£ŘèąŇåĹĘåÿČåijŔäžžåŁąïijŇæŐĺè■

ä¡ăäź§åŔŕäžěä¡£çŤĺ `python -m paddle.distributed.launch` æİěåŘŕåŁĺèő■çżČäžžåŁąïijŇäžŇåőđäÿŁïijŇ `fleetrun`æŸŕåĽ■èĂĚçŽĎå£ńæ■ůæŰźåijŔãĆĆ

5.3.1 éŻĘåŘĹéĂŽä£ąèő■çżČ

- **GPUå■ŢæIJžå■Ţå■ąèő■çżČ**

  å■ŢæIJžå■Ţå■ąæIJĽäÿď çğ■æŰźåijŔïijŽäÿĂçğ■åŘŕçŻť æŐěä¡£çŤĺpythonæĽğèąŇïijŇäź§åŘŕäžěä¡£çŤ

  ãĂŘæŰźæşŢäÿĂãĂŚçŻť æŐěä¡£çŤĺpythonæĽğèąŇ

```
export CUDA_VISIBLE_DEVICES=0
python train.py
```

  ãĂŘæŰźæşŢäžŇãĂŚä¡£çŤĺfleetrunæĽğèąŇ

```
fleetrun --gpus=0 train.py
```

  æşĺïijŽåęĆæđIJæŇĞåőŽäžĘexport CUDA_VISIBLE_DEVICES=0 ïijŇåĹŹåŘŕäžěçŻť æŐěä¡£çŤĺïijŽ

```
export CUDA_VISIBLE_DEVICES=0
fleetrun train.py
```

- **GPUå■ŢæIJžåď Žå■ąèő■çżČ**

  èŃěåŘŕåŁĺå■ŢæIJž4å■ąçŽĎäžžåŁąïijŇåŘłéIJĂéĂŽè£Ğ--gpusæŇĞåőŽçĺ'žéŮšçŽĎ4åijäå■ąå■şåŘŕãĂ

```
fleetrun --gpus=0,1,2,3 train.py
```

  æşĺïijŽåęĆæđIJæŇĞåőŽäžĘexport CUDA_VISIBLE_DEVICES=0,1,2,3ïijŇåĹŹåŘŕäžěçŻť æŐěä¡£çŤĺïijŽ

```
export CUDA_VISIBLE_DEVICES=0,1,2,3
fleetrun train.py
```

- **GPUåď ŽæIJžåď Žå■ąèő■çżČ**

  [çď'žä¿ŃäÿĂ] 2æIJž8å■ą (æŕŔäÿłèŁĆçĆź4å■ą)

```
fleetrun --ips="xx.xx.xx.xx,yy.yy.yy.yy" --gpus=0,1,2,3 train.py
```

  æşĺïijŽåęĆæđIJæŕŔåŘřæIJžåŹĺåİĞæŇĞåőŽäžĘexport CUDA_VISIBLE_DEVICES=0,1,2,3ïijŇåĹŹåŘŕäžěçŻť æŐěåIJĺæŕŔåŘřèŁĆçĆźäÿŁåŘŕåŁĺïijŽ

```
export CUDA_VISIBLE_DEVICES=0,1,2,3
fleetrun --ips="xx.xx.xx.xx,yy.yy.yy.yy" train.py
```

[çd'žä¿ŃäžŇ] 2æIJž16å■ąïijĹæŕŔäÿłèŁĆçĆź8å■ąïijŇåĄĜèő¿æŕŔåŔřæIJžåŹĺåİĞæIJL'8å■ąåŔŕä¡£çŤĺïijĺ

```
fleetrun --ips="xx.xx.xx.xx,yy.yy.yy.yy" train.py
```

5.3.2 åŔĆæŢřælJ■åŁąåŹĺèő■çżČ

åIJÍCPUéŻĘç¿d'è£ŘèąŇåŔĆæŢřælJ■åŁąåŹÍ

- **åŔĆæŢřæIJ■åŁąåŹĺèő■çżČ - å■ŢæIJžæĺąæŃ§åĹĘåÿČåijŔèő■çżČ**

1åŘřæIJžåŹĺéĂŽè£ĜåďŽè£ŻçÍŃæĺąæŃ§åĹĘåÿČåijŔèő■çżČïijŇ1äÿłæIJ■åŁąèŁĆçĆźæŘ■éĚ■4äÿłèő■çż

fleetrunåŘŕåŁĺæŮűåŔłéIJĂæŇĞåőŽæIJ■åŁąèŁĆçĆźæŢř--server_numåŠŇèő■çżČèŁĆçĆźæŢř-

```
fleetrun --server_num=1 --worker_num=4 train.py
```

- **åŔĆæŢřæIJ■åŁąåŹĺèő■çżČ - èĞłåőŽäźL'åď'ŽæIJžèő■çżČ**

fleetrunåŘŕåŁĺæŮűåŔłéIJĂæŇĞåőŽæIJ■åŁąèŁĆçĆźçŽDipåŠŇçńŕåŘčåĹŮèąÍ--servers åŠŇèő■çżČèŁĆçĆźçŽDipåĹŮèąÍ--workers ïijŇå■şåŔŕè£ŻèąŇåď'ŽæIJžèő■çżČãĂĆ äÿŃåĹŮçď'žä¿Ńäÿ■ïijŇxx.xx.xx.xxäżčèąĺæIJžåŹĺ1ïijŇyy.yy.yy.yyäżčèąĺæIJžåŹĺ2ïijŇ6170äżčèąÍçŤĺæĹůæŇČ

```
# 2äÿłservers 8äÿłworkers
fleetrun --servers="xx.xx.xx.xx:6170,yy.yy.yy.yy:6171" --workers=
↪"xx.xx.xx.xx,xx.xx.xx.xx,xx.xx.xx.xx,xx.xx.xx.xx,yy.yy.yy.yy,yy.
↪yy.yy.yy,yy.yy.yy.yy,yy.yy.yy.yy" train.py
```

--workersåŔĆæŢřåŕŕäżěäżĚæŇĞåőŽipåĹŮèąÍïijŇæ■ď'æŮűfleetrunårĘäijŽåIJÍåŘŕåŁĺèő■çżČäż

```
# 2äÿłservers 8äÿłworkers
fleetrun --servers="xx.xx.xx.xx:6170,yy.yy.yy.yy:6171" --workers=
↪"xx.xx.xx.xx:6172,xx.xx.xx.xx:6173,xx.xx.xx.xx:6174,xx.xx.xx.
↪xx:6175,yy.yy.yy.yy:6176,yy.yy.yy.yy:6177,yy.yy.yy.yy:6178,yy.yy.
↪yy.yy:6179" train.py
```

åIJÍGPUéŻĘç¿d'è£ŘèąŇåŔĆæŢřælJ■åŁąåŹÍ

- **åŔĆæŢřæIJ■åŁąåŹĺèő■çżČ - å■ŢæIJžæĺąæŃ§åĹĘåÿČåijŔèő■çżČ**

1åŘřæIJžåŹĺéĂŽè£ĜåďŽè£ŻçÍŃæĺąæŃ§ïijŇ2äÿłæIJ■åŁąèŁĆçĆźæŘ■éĚ■4äÿłèő■çżČèŁĆçĆźïijŇæŕŔäÿ

```
# 2äÿłserver 4äÿłworker
export CUDA_VISIBLE_DEVICES=0,1,2,3
fleetrun --server_num=2 --worker_num=4 train.py
```

1åŘřæIJžåŹĺéĂŽè£ĜåďŽè£ŻçÍŃæĺąæŃ§ïijŇ 2äÿłæIJ■åŁąèŁĆçĆźæŘ■éĚ■2äÿłèő■çżČèŁĆçĆźïijŇäÿď'ä

```
# 2äÿłserver 2äÿłworker
export CUDA_VISIBLE_DEVICES=0
fleetrun --server_num=2 --worker_num=2 train.py
```

- **åŔĆæŢřæIJ■åŁąåŹĺèő■çżČ - èĞłåőŽäźL'åd'ŽæIJžèő■çżČ**

fleetrunåŘŕåŁĺæŮűåŔłéIJĂæŇĞåőŽæIJ■åŁąèŁĆçĆźçŽĎipåŠŇçńŕåŘčåĹŮèąĺ--servers åŠŇ èő■çżČèŁĆçĆźçŽĎipåŠŇçńŕåŘčåĹŮèąĺ--workers ïijŇå■şåŔŕè£ŻèąŇåd'ŽæIJžèő■çżČãĂĆ

äžěäÿŃçd'žä¿Ńäÿ■ïijŇxx.xx.xx.xxäžčèąĺæIJžåŹĺ1ïijŇyy.yy.yy.yyäžčèąĺæIJžåŹĺ2ïijŇ6170äžčèąĺçŤĺæĹůa

```
#␣
↪2åŘŕæIJžåŹĺïijŇæŕŔåŘŕæIJžåŹĺåİĞæIJL'1äÿłæIJ■åŁąèŁĆçĆźïijŇ1äÿłèő■çżČèŁĆçĆź
# 2äÿłserver 2äÿłworker
# æŕŔåŘŕæIJžåŹĺåİĞæŇĞåőŽäžĘåŔŕçŤĺèő¿åd'Ğ GPU:0
export CUDA_VISIBLE_DEVICES=0
fleetrun --servers="xx.xx.xx.xx:6170,yy.yy.yy.yy:6171" --workers=
↪"xx.xx.xx.xx:6172,yy.yy.yy.yy:6173" train.py
```

äžěäÿŃçd'žä¿Ńäÿ■ïijŇfleetrunåřĘåĹĘåĹńåIJĺ2åŘŕæIJžåŹĺäÿŁåŘŕåŁĺ1äÿłæIJ■åŁąèŁĆçĆźïijŇ4äÿłèő

```
#␣
↪2åŘŕæIJžåŹĺïijŇæŕŔåŘŕæIJžåŹĺåİĞæIJL'1äÿłæIJ■åŁąèŁĆçĆźïijŇ4äÿłèő■çżČèŁĆçĆź
# 2äÿłserver 4äÿłworker
# æŕŔåŘŕæIJžåŹĺåİĞæŇĞåőŽäžĘåŔŕçŤĺèő¿åd'Ğ GPU:0,1,2,3
export CUDA_VISIBLE_DEVICES=0,1,2,3
fleetrun --servers="xx.xx.xx.xx:6170,yy.yy.yy.yy:6171" --workers=
↪"xx.xx.xx.xx:6172,xx.xx.xx.xx:6173,xx.xx.xx.xx:6174,xx.xx.xx.
↪xx:6175,yy.yy.yy.yy:6176,yy.yy.yy.yy:6177,yy.yy.yy.yy:6178,yy.yy.
↪yy.yy:6179" train.py
```

åijĆædĎéŻEç¿d'è£ŘèąŇåŔĆæŢřæIJ■åŁąåŹĺ

- **åŔĆæŢřæIJ■åŁąåŹĺèő■çżČ - å■ŢæIJžæĺąæŃ§åĹĘåÿČåijŔèő■çżČ**

1åŘŕæIJžåŹĺéĂŽè£Ğåd'Žè£ŻçĺŃæĺąæŃ§ïijŇ2äÿłæIJ■åŁąèŁĆçĆźæŘ■éĚ■2äÿłèő■çżČèŁĆçĆźäžěåŔŁ2ä

```
# 2äÿłserver 4äÿłworker
export CUDA_VISIBLE_DEVICES=0,1
fleetrun --server_num=2 --worker_num=2 --heter_worker_num=2 train.py
```

5.4 fleetrunåŚ¡äžd'åŔĆæŢřäžŃçż■

- CollectiveæĺąåijŔçŻÿåĚşåŔĆæŢř:
  - ips ïijĹstrïijŇåŘŕéĂL'ïijL'ïijŽ æŇĞåőŽéĂL'æŃľ'åŞłäžŻèŁĆçĆźIPè£ŻèąŇèő■çżČïijŇéžŸèőd'äÿžã å■şäijŽåIJĺæIJňåIJŕæL'ğèąŇå■ŢæIJžå■Ţå■ąæĹŰåd'Žå■ąèő■çżČãĂĆ

  - gpusïijĹstr, åŕŕéĂL'ïijL'ïijŽæŇĞåőŽéĂL'æŇľåŞłäžŻGPUå■ąè£ŻèąŇèő■çżČïijŇézŸèőd'äÿžNoneïijŇå■şäijŽéĂL'æŇľ CUDA_VISIBLE_DEVICESæL'ĂæŸ¿çd'žçŽĎæL'ĂæIJL'å■ąãĂĆäÿ■èő¿ç¡ő nproc_per_nodeåŔĆæŢřæŮűïijŇářĘåŕŕåŁĺGPUäÿłæŢřäÿłè£ŻçĺŇè£ŻèąŇèő■çżČïijŇæŕŔäÿłè
  - nproc_per_node ïijĹint, åŕŕéĂL'ïijL'ïijŽèő¿ç¡őåd'ŽářŚäÿłè£ŻçĺŇè£ŻèąŇèő■çżČãĂĆèő¿ç¡őæŢřçŻ

- åŔĆæŢřæIJ■åŁąåŹĺæĺąåijŔåŕŕéĚ■åŔĆæŢř:
  - server_numïijĹintïijŇåŕŕéĂL'ïijL'ïijŽå■ŢæIJžæĺąæŃ§åĹĘåÿČåijŔäżżåŁąäÿ■ïijŇæŇĞåőŽåŔĆæŢ
  - worker_numïijĹintïijŇåŕŕéĂL'ïijL'ïijŽå■ŢæIJžæĺąæŃ§åĹĘåÿČåijŔäżżåŁąäÿ■ïijŇæŇĞåőŽåŔĆæ
  - heter_worker_numïijĹintïijŇåŕŕéĂL'ïijL'ïijŽåIJĺåijĆæđĎéŻEç¿d'äÿ■åŕŕåŁĺå■ŢæIJžæĺąæŃ§åĹĘå
  - serversïijĹstr, åŕŕéĂL'ïijL'ïijŽ åd'ŽæIJžåĹĘåÿČåijŔäżżåŁąäÿ■ïijŇæŇĞåőŽåŔĆæŢřæIJ■åŁąåŹĺæ
  - workersïijĹstr, åŕŕéĂL'ïijL'ïijŽ åd'ŽæIJžåĹĘåÿČåijŔäżżåŁąäÿ■ïijŇæŇĞåőŽåŔĆæŢřæIJ■åŁąåŹĺè
  - heter_workersïijĹstr, åŕŕéĂL'ïijL': åIJĺåijĆæđĎéŻEç¿d'äÿ■åŕŕåŁĺåĹĘåÿČåijŔäżżåŁąïijŇæŇĞåőŽ
  - http_portïijĹint, åŕŕéĂL'ïijL'ïijŽåŔĆæŢřæIJ■åŁąåŹĺæĺąåijŔäÿ■ïijŇçŤĺ-GlooåŕŕåŁĺæŮűèő¿ç¡őçŽĎè£đæŐěçńŕåŔč

- åĚűäżŰïijŽ
  - log_dirïijĹstr, åŕŕéĂL'ïijL'ïijŽ æŇĞåőŽåĹĘåÿČåijŔäżżåŁąèő■çżČæŮěå£ŮçŽĎä£Ìå■Ÿèůŕå¿ĎïijŇ

5.5 ä¡£çŤĺfleetrunè£ŻèąŇGPUåd'Žå■ąèő■çżČåőđä¿Ń

äÿŃéÌćæĹŚäžňářĘéĂŽè£Ğä¿Ńå■ŘïijŇäÿžæĆĺèŕęçżĘäżŃçż■åęĆä¡ŢåĹ'çŤĺfleetrunåřĘå■ŢæIJžå■Ţå
è£ŹéĞŇä¡£çŤĺäÿŐéÌŹæĂĄåŽ¿åĹĘåÿČåijŔèő■çżČå£ńéĂ§åijĂåğŃ
çŻÿåŘŇçŽĎçd'žä¿Näžčçăąè£ŻèąŇèŕť'æŸŐãĂĆ

```
import os
import time
import paddle
import paddle.distributed.fleet as fleet
import paddle.static.nn as nn
import paddle.fluid as fluid

def mnist_on_mlp_model():
    train_dataset = paddle.vision.datasets.MNIST(mode='train')
    test_dataset = paddle.vision.datasets.MNIST(mode='test')
    x = paddle.data(name="x", shape=[64, 1, 28, 28], dtype='float32
↪')
    y = paddle.data(name="y", shape=[64, 1], dtype='int64')
    x_flatten = fluid.layers.reshape(x, [64, 784])
    fc_1 = nn.fc(input=x_flatten, size=128, act='tanh')
    fc_2 = nn.fc(input=fc_1, size=128, act='tanh')
    prediction = nn.fc(input=[fc_2], size=10, act='softmax')
    cost = fluid.layers.cross_entropy(input=prediction, label=y)
```

(äÿŃéąçżğçż■)

(çż■äÿŁéąţ)

```
    acc_top1 = fluid.layers.accuracy(input=prediction, label=y, k=1)
    avg_cost = fluid.layers.mean(x=cost)
    return train_dataset, test_dataset, x, y, avg_cost, acc_top1

paddle.enable_static()
train_data, test_data, x, y, cost, acc = mnist_on_mlp_model()
place = paddle.CUDAPlace(int(os.environ.get('FLAGS_selected_gpus',␣
↪0)))
train_dataloader = paddle.io.DataLoader(
    train_data, feed_list=[x, y], drop_last=True,
    places=place, batch_size=64, shuffle=True)
fleet.init(is_collective=True)
strategy = fleet.DistributedStrategy()
#optimizer = paddle.optimizer.Adam(learning_rate=0.01)
optimizer = fluid.optimizer.Adam(learning_rate=0.001)
optimizer = fleet.distributed_optimizer(optimizer,␣
↪strategy=strategy)
optimizer.minimize(cost)

exe = paddle.static.Executor(place)
exe.run(paddle.static.default_startup_program())

epoch = 10
for i in range(epoch):
    total_time = 0
    step = 0
    for data in train_dataloader():
        step += 1
        start_time = time.time()
        loss_val, acc_val = exe.run(
          paddle.static.default_main_program(),
          feed=data, fetch_list=[cost.name, acc.name])
        if step % 200 == 0:
            end_time = time.time()
            total_time += (end_time - start_time)
            print(
                    "epoch: %d, step:%d, train_loss: %f, total time␣
↪cost = %f, speed: %f"
                % (i, step, loss_val[0], total_time,
                   1 / (end_time - start_time) ))
```

5.5.1 å■ŢælJžå■Ţå■ąèő■çżČ

åřĘäÿŁè£řäžčçăĄä£Ĺå■ŸåIJĺtrain.pyäžčçăĄäÿ■ïijŇå■ŢæIJžå■Ţå■ąèő■çżČå■ĄåĹĘçŽĎçőĂå■Ţïij

```
export CUDA_VISIBLE_DEVICES=0
python train.py
```

åŔŕäžěçIJŃèğĄçżĹçńŕäÿŁæĽ'Şå■řæŮěå£Ůä£ąæĄŕïijŽ

```
  epoch: 0, step:200, train_loss: 0.424425, total time cost = 0.
↪000947, speed: 1055.967774
  epoch: 0, step:400, train_loss: 0.273742, total time cost = 0.
↪001725, speed: 1285.413423
  epoch: 0, step:600, train_loss: 0.472131, total time cost = 0.
↪002467, speed: 1347.784062
  epoch: 0, step:800, train_loss: 0.445613, total time cost = 0.
↪003184, speed: 1394.382979
  epoch: 1, step:200, train_loss: 0.512807, total time cost = 0.
↪000681, speed: 1468.593838
  epoch: 1, step:400, train_loss: 0.571385, total time cost = 0.
↪001344, speed: 1508.199928
  epoch: 1, step:600, train_loss: 0.617232, total time cost = 0.
↪002034, speed: 1449.310297
  epoch: 1, step:800, train_loss: 0.392537, total time cost = 0.
↪002813, speed: 1283.446756
  epoch: 2, step:200, train_loss: 0.288508, total time cost = 0.
↪000796, speed: 1256.155735
  epoch: 2, step:400, train_loss: 0.448433, total time cost = 0.
↪001531, speed: 1360.461888
  epoch: 2, step:600, train_loss: 0.593330, total time cost = 0.
↪002292, speed: 1314.005013
...
```

5.5.2 å■ŢælJžåď'Žå■ąèő■çżČ

äžŐå■ŢæIJžå■Ţå■ąèő■çżČåĹřå■ŢæIJžåď'Žå■ąèő■çżČäÿ■éIJĂèęĄæŤźåŁĺtrain.pyäžčçăĄïijŇåŔłéIJĂæŤźäÿĂèąŇåŘŕåŁĺåŚ¡äžď'ïijŽ

```
export CUDA_VISIBLE_DEVICES=0,1,2,3
fleetrun train.py
```

èő■çżČæŮěå£ŮåŔŕäžěåIJĺçżĹçńŕäÿŁæ§ěçIJŃïijŇäź§åŘŕçĺ■åŘŐåIJĺ./log/çŽőå¡ŢäÿŃæ§ěçIJŃæŕŔäÿłå■ąçżĹçńŕåŔŕäžěçIJŃåĹřæŸ¿çď'žæŮěå£ŮåęĆäÿŃïijŽ

```
-----------  Configuration Arguments -----------
gpus: 0,1,2,3
ips: 127.0.0.1
log_dir: log
server_num: None
servers:
training_script: train.py
training_script_args: []
worker_num: None
workers:
------------------------------------------------
INFO 202X-0X-0X 06:09:36,185 launch_utils.py:425] Local start 4␣
↪processes. First process distributed environment info (Only ÿÑæçżěçż■)
↪Debug):
```

(çż■äÿŁéąţ)

```
=============================================================================
            Distributed Envs                Value
-----------------------------------------------------------------------------
 ↪-------------------
PADDLE_CURRENT_ENDPOINT                     127.0.0.1:33360
PADDLE_TRAINERS_NUM                         4
FLAGS_selected_gpus                         0
PADDLE_TRAINER_ENDPOINTS                    ... 0.1:11330,127.0.0.
 ↪1:54803,127.0.0.1:49294
PADDLE_TRAINER_ID                           0
=============================================================================
 epoch: 0, step:200, train_loss: 0.306129, total time cost = 0.
 ↪001170, speed: 854.759323
 epoch: 0, step:400, train_loss: 0.287594, total time cost = 0.
 ↪002226, speed: 947.009257
 epoch: 0, step:600, train_loss: 0.179934, total time cost = 0.
 ↪003201, speed: 1025.752996
 epoch: 0, step:800, train_loss: 0.137214, total time cost = 0.
 ↪005004, speed: 554.582044
 epoch: 1, step:200, train_loss: 0.302534, total time cost = 0.
 ↪000975, speed: 1025.752996
 epoch: 1, step:400, train_loss: 0.375780, total time cost = 0.
 ↪001934, speed: 1042.581158
 epoch: 1, step:600, train_loss: 0.247651, total time cost = 0.
 ↪002892, speed: 1043.878547
 epoch: 1, step:800, train_loss: 0.086278, total time cost = 0.
 ↪003845, speed: 1049.363022
.....
```

CHAPTER 6

FleetXå£ńéĂ§åijĂåğŃ

6.1 FleetXæŸŕäżĂäźĹïij§

`FleetX`æŔŘä¿ŻæŢĹçŐĞæIJĂéńŸçŽĎåĹĘåÿČåijŔæĺąåđŃéćĎèő■çżČåŁ§èČ¡ïijŇåőČåŔŕäżěä¡IJäÿžpa
`distributed.fleet`çŽĎæL'ľåśŢè£ŻèąŇéĚ■åŘĹä¡£çŤĺãĂĆ

6.2 æŔŘä¿ŻåŞłäžŻåŁ§èČ¡ïij§

- ç§■äżčçăĄåőŽäźL'éćĎèő■çżČæĺąåđŃ
- éćĎç¡őçżŔåĚÿæĺąåđŃçŽĎåĚňåijĂèő■çżČæŢřæ■ő
- çŤĺæĹůåŔŕä¡ŐæĹŘæIJňæŻ£æ■ćèĞłæIJL'æŢřæ■őéŻĘ
- éİćåŘŚærŔäÿłæĺąåđŃçŽĎæIJĂä¡şåĹĘåÿČåijŔèő■çżČåőđèůţ

6.3 äÿŁæL'Ńçď'žä¿Ń

äżěäÿŃéĂŽè£ĞåŻ¿åČŔåĹĘçśżResnet50çŽĎä¿Ńå■ŘïijŇèŕť'æŸŐåęĆä¡Ţä¡£çŤĺFleetXçŽĎæŐěåŔčè£Żè

- åŕijåĚěä¿İèţŰ
- æđĎåżžæĺąåđŃ
- åőŽäźL'åĹĘåÿČåijŔç■ŰçŢě
- åijĂåğŃèő■çżČ

äÿžäžĘçőĂåŇŰæĺąåđŃåőŽäźL'çŽĎè£ĞçĺŃïijŇæĹŚäżňåIJĺåŘŐéİćçŽĎæŰĞæąčäÿ■äijŽår¡éĞŔä¡£çŤĺFl

6.3.1 1. åŕijåĚëä¿ièţŰ

FleetXä¿ÌèţŰPaddle 1.8.0åŔŁäźŇåŘŐçŽĎçL'ŁæIJñãĂĆèŕůçąőèőd'åůšåőL'èčĚæ■čçąőçŽĎ-PaddleçL'ŁæIJňïijŇåźűæŇL'çĚgäžëäÿŇæŰźåijŔåŕijåĚëPaddle åŔŁ FleetXãĂĆ

```
import paddle
import paddle.distributed.fleet as fleet
import fleetx as X
```

6.3.2 2. æđĎåżžæÍąåđŃ

éĂŽè£ĞFleetXæŔŘä¿ŻçŽĎ `X.applications` æŐěåŔčïijŇçŤÍæĹůåŔŕäžëä¡£çŤÍäÿĂèąŇäžčçăĄåŁãè

```
paddle.enable_static()
configs = X.parse_train_configs()

model = X.applications.Resnet50()
downloader = X.utils.Downloader()
local_path = downloader.download_from_bos(fs_yml="https://xxx.xx.xx.
↪xx/full_imagenet_bos.yml", local_path='./data')
loader = model.get_train_dataloader(local_path, batch_size=32)
```

6.3.3 3. åőŽäźL'åĹĘåÿČåijŔç■ŰçŢě

åIJÍåőŽäźL'åőŇå■ŢæIJžèő■çżČç¡ŚçżIJåŘŐïijŇçŤÍæĹůåŔŕäžëä¡£çŤÍ`paddle.distributed.fleet.DistributedStrategy()` æŐěåŔčåőŽäźL'åĹĘåÿČåijŔç■ŰçŢěïijŇåřĘæÍąåđŃ

```
# ä¡£çŤÍpaddle.distributed.fleetè£ŻèąŇcollective training
fleet.init(is_collective=True)
# åőŽäźL'DistributedStrategy
dist_strategy = fleet.DistributedStrategy()
# èčĚéěřå■ŢæIJžoptimizeräÿžåĹĘåÿČåijŔoptimizer
optimizer = paddle.distributed.fleet.distributed_
↪optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)
```

6.3.4 4. åijĂågŃèő■çżČ

åŔŕäžëä¡£çŤÍ`FleetX`åĘĚç¡őçŽĎèő■çżČåŹÍè£ŻèąŇå£ńéĂ§èő■çżČïijŇæŰźä¿£çőŮæşŢåůěçÍŃåÿĹå£ńé

```
trainer = X.MultiGPUTrainer()
trainer.fit(model, loader, epoch=10)
```

çŤÍæĹůäź§åŔŕäžëéĞĞçŤÍPaddleåŐ§çŤ§çŽĎAPIè£ŻèąŇèő■çżČætĄçÍŃçŽĎåőŽäźL'ïijŇäžčçăĄåęĆäÿŃ

```
exe = fluid.Executor(place)
exe.run(fluid.default_startup_program())

for epoch_id in range(5):
    step_id = 0
    for data in loader:
        cost_val = exe.run(paddle.default_main_program(),
                           feed=data,
                           fetch_list=[model.loss.name])
        if step_id % 100 == 0:
            print("worker index: %d, epoch: %d, step: %d, train␣
↪loss: %f"
                  % (fleet.worker_index(), epoch_id, step_id, cost_
↪val[0]))
```

äżŐPaddle 2.0 rccL'ĹæIJňåijĂåğŃïijŇæĹŚäžňçż§äÿĂéĞĞçŤĺfleetrunè£ŻèąŇåd'Žå■ąèő■çżČçŽĎåŘŕåŁĺ

```
fleetrun --gpus 0,1,2,3 resnet_app.py
```

åĚşäžŐfleetrunåŚ¡äżd'ïijŇæŻt'èŕęçżĘçŽĎä¡£çŤĺèŕt'æŸŐèŕůåŔĆèĂČfleetrun

CHAPTER 7

ä¡£çŤÍfleetsubæŔŘäžďéŻĘç¿ďäżżåŁą

7.1 fleetsubæŸŕäżĂäźĹ

å¡ŞæĆÍåőL'èčĚäžĘfleet-xåŘŐïijŇä¿£åŔŕäżěä¡£çŤÍfleetsubåIJÍéŻĘç¿ď'äÿŁæŔŘäžď'åĹĘåÿČåijŔä

7.2 ä¡£çŤÍèęĄæśĆ

ä¡£çŤÍfleetsubåŚ¡äżď'çŽĎèęĄæśĆïijŽåőL'èčĚfleet-x

- ãĂŘæŰźæşŢäÿĂãĂŚäżŐpipæžŘåőL'èčĚ

```
pip install fleet-x
```

- ãĂŘæŰźæşŢäžŇãĂŚäÿŃè¡¡whlåŇĚåźűåIJÍæIJňåIJřåőL'èčĚ

```
# python2
wget --no-check-certificate https://fleet.bj.bcebos.com/fleet_x-0.0.
↪4-py2-none-any.whl
pip install fleet_x-0.0.4-py2-none-any.whl
# python3
wget --no-check-certificate https://fleet.bj.bcebos.com/fleet_x-0.0.
↪4-py3-none-any.whl
pip3 install fleet_x-0.0.4-py3-none-any.whl
```

7.3 ä¡£çŤĺèŕťæŸŐ

åIJÍæŔŘäžďäżżåŁąåĽ■ïijŇçŤÍæĹůéIJĂèęĄåIJÍyamlæŰĞäžűäÿ■éĚ■ç¡őäżżåŁąçŻÿåĚşçŽĎä£ąæĄŕïijŇåę

éęŰåĚĹçIJŃäÿĂäÿłyamlæŰĞäžűçŽĎæăůä¿ŃãĂĆåŻääÿžä£ąæĄŕåőĽåĚĺçŽĎåŐ§åŻăyamlæŰĞäžűäÿ■çŽ

```
num_trainers: 4
num_cards: 8
job_prefix: bert_base_pretraining
image_addr: ${image_addr:-"dockhub.com/paddlepaddle-public/paddle_
↪ubuntu1604:cuda10.0-cudnn7-dev"}
cluster_name: v100-32-cluster
group_name: k8s-gpu-v100-8
server: paddlecloud.server.com
log_fs_name: "afs://xx.fs.com:9902"
log_fs_ugi: "ugi_name,ugi_passwd"
log_output_path: "/xx/yy/zz"
file_dir: "./"

whl_install_commands:
  - pip install fleet_x-0.0.5-py2-none-any.whl
  - pip install paddlepaddle_gpu-0.0.0-cp27-cp27mu-linux_x86_64.whl

commands:
  - fleetrun bert_base.py --download_config=bert.yaml
```

å■Ůæőţåř■çğř	å■ŮæőţåŘńäźĽ	çśžåđŃ
num_trainers	èő■çżČèŁĆçĆźçŽĎæŢřéĞŔ	INT
num_cards	å■ŢèŁĆçĆźäÿŁçŤşèřůçŽĎGPUå■ąæŢř	INT
job_prefix	äżżåŁąåŘ■åĽ■çijĂ	STRING
image_addr	éŢIJåČŔåIJřåİĂ	{STRING}
cluster_name	éŻEç¿d'åŘ■	STRING
group_name	ç¿d'çżĎåŘ■	STRING
server	éŻEç¿d'masterèŁĆçĆźæIJ■åŁąåŘ■	STRING
log_fs_name	äżżåŁąæŮěå£Ůå■ŸæŤ¿çŽĎæŰĞäžűçşżçż§åŘ■	STRING
log_fs_ugi	äżżåŁąæŮěå£Ůå■ŸæŤ¿çŽĎæŰĞäžűçşżçż§UGI	STRING
log_output_path	äżżåŁąæŮěå£Ůå■ŸæŤ¿çŽĎçŻőæăĞæŰĞäžűçşżçż§åŘ■ĺĂ	STRING
file_dir	æŔŘäžďäżżåŁąéIJĂèęĄäÿŁäijăçŽĎæŰĞäžűçŻőåŢ	STRING
whl_install_commands	åśĽèčĚåŘĎçğ■wheelåŇĚçŽĎåŚ¡äžď	Repeated Command Line
commands	è£ŘèąŇäżżåŁąæĽğèąŇçŽĎåŘĎçğ■åŚ¡äžď	Repeated Command Line

7.4 äżżåŁąæŔŘäžď

åőŽäźĽ'åőŇäÿŁè£řèĎŽæIJňåŘŐïijŇçŤĺæĹůå■şåŔŕä¡£çŤĺ`fleetsub`åŠ¡äżď'åŘŚPaddleCloud æŔŘäžď'äżżåŁąäžĘïijŽ

```
fleetsub -f demo.yaml
```

7.5 ä¡£çŤĺæăůä¿Ń

åĚůä¡ŞçŽĎä¡£çŤĺèŕť'æŸŐåŔŁæăůä¿ŃäžčçăĄèŕůåŔĆèĂČäÿŃéİćçŽĎWIKI

CHAPTER 8

ä¡£çŤÍFleetè£Żèą ŇåŔĆæŢřæIJ■åŁąåŹÍèő■çżČ

åIJÍåď ğæŢřæ■őæţłæ¡őçŽĎæŐÍåŁÍäÿŃïijŇæIJL'æăĞç■¿èő■çżČæŢřæ■őçŽĎèğĎæÍąåŔŰå¿ŮäžĘéčđéĂ§ç
Speech 2çşżçż§ä¡£çŤÍäžĘ11940åřŔæŮűçŽĎèŕ■é§şæŢřæ■őäžěåŔŁèűÉè£Ğ200äÿĞåŔěèąÍè£řæİěèő■çżČèŕ■é§

äÿžäžĘæŔŘéńŸæÍąåđŇçŽĎèő■çżČæŢĹçŐĞïijŇåĹĘåÿČåijŔèő■çżČåžŤè£ŘèĂŇçŤ§ïijŇåĚűäÿ■å§žäžŐ

- **èő■çżČèŁĆçĆź**ïijŽèŕěèŁĆçĆźèť §èť'čåőŇæĹŘæŢřæ■őèŕżåŔŰãĂĄåL'■åŘŚèőąçőŮãĂĄåŔ■åŘŚæćŕåžę
- **æIJ■åŁąèŁĆçĆź**ïijŽåIJÍæŤűåĹřæL'ĂæIJL'èő■çżČèŁĆçĆźäijăæİěçŽĎæćŕåžęåŘŐïijŃèŕěèŁĆçĆźäijŽå

æăźæ■őåŔĆæŢřæŻť'æŰřçŽĎæŰźåijŔäÿ■åŘŇïijŇåŔŕäžěåĹĘäÿžåŘŇæ■ěåŠŇåijĆæ■ěäÿď'çğ■ïijŽ

- **åŘŇæ■ěèő■çżČ**ïijŽåIJÍåŘŇæ■ěåŔĆæŢřæIJ■åŁąåŹÍåĹĘåÿČåijŔèő■çżČäÿ■ïijŇæL'ĂæIJL'èő■çżČèŁĆç
- **åijĆæ■ěèő■çżČ**ïijŽäÿŐåŘŇæ■ěèő■çżČäÿ■åŘŇïijŇåIJÍåijĆæ■ěèő■çżČäÿ■äżżä¡Ţäÿď'äÿłèő■çżČèŁĆçĆ

äÿŃéİćæĹŚäžňårĘéĂŽè£Ğä¿Ńå■ŘïijŇäÿžæĆÍäżŃçż■åŘŇæ■ě/åijĆæ■ěèő■çżČåIJÍFleetäÿ■çŽĎåőđçŐřã

åIJÍåijĂåğŃäźŃåL'■æĹŚäžňéęŰåĚĹéIJĂèęĄäÿŃè¡¡èő■çżČäÿ■æL'ĂéIJĂèęĄçŽĎæŢřæ■őïijŽ

```
# äÿŃè¡¡åźűèğčåŐŃæŢřæ■őïijŇèő■çżČæŢřæ■őèőšä£Íå■ŸèĞşåŘ■äÿž raw_data␣
↪çŽĎæŰĞäżűåd'ź
wget --no-check-certificate https://fleet.bj.bcebos.com/ctr_data.
↪tar.gz
tar -zxvf ctr_data.tar.gz
```

8.1 åőđçŤÍæăůä¿Ń

äÿŃéİćæĹŚäžňæİěäžŃçż■åęĆä¡ŢçŤÍFleetæŐěåŔčïijŇåőŇæĹŘåŔĆæŢřæIJ■åŁąåŹÍåĹĘåÿČåijŔèő■çżČï

8.1.1 åŕijåĚëä¿ìèţŰ

```
import paddle
import os
import fleetx as X
import paddle.fluid as fluid
import paddle.distributed.fleet as fleet
import paddle.distributed.fleet.base.role_maker as role_maker
```

8.1.2 åőŽäźĽåĹĘåÿČåijŕæĺąåijŕåźűåĹìåğŃåŇŰ

éĂŽè£ĞX.parse_train_configs()æŐěåŕčïijŇçŤĺæĹůåŕŕäžěåőŽäźĽ'èő■çżČçŻÿåĚşçŽĎåŔĆæŢ
init()æŐěåŕčåőŽäźĽ'äžĘåĹĘåÿČåijŕæĺąåđŃïijŇinit()æŐěåŕčéżŸèőd'ä¡£çŤĺåŔĆæŢřæIJ■åŁąåŹĺæĺą
äźĽ'äżżä¡ŢåŔĆæŢřãĂĆ

```
paddle.enable_static()
configs = X.parse_train_configs()
role = role_maker.PaddleCloudRoleMaker()
fleet.init(role)
```

8.1.3 åŁăè¡¡æĺąåđŃåŔŁæŢřæ■ő

çŤĺæĹůåŕŕäžěéĂŽè£ĞX.applicationsæŐěåŕčåŁăè¡¡æĹŚäžňéćĎåĚĹåőŽäźĽ'åě¡çŽĎæĺąåđŃãĂĆå
DNNæĺąåđŃïijŇåŘŇæŮűçŤĺæĹůåŕŕäžěäÿžæĺąåđŃåőŽåĹűçŽĎdata_loaderæŐěåŕčåŁăè¡¡æŢřæ■ő.

```
model = X.applications.MultiSlotCTR()
loader = model.load_criteo_from_file('./train_data')
```

8.1.4 åőŽäźĽåŘŇæ■ěèő■çżČ Strategy åŔŁ Optimizer

åIJĺFleet APIäÿ■ïijŇçŤĺæĹůåŕŕäžěä¡£çŤĺfleet.DistributedStrategy()æŐěåŕčåőŽäźĽ'èĞłåů

åĚűäÿ■a_syncéĂL'éąźçŤĺäžŐåőŽäźĽ'åŔĆæŢřæIJ■åŁąåŹĺçŻÿåĚşçŽĎç■ŰçŢěïijŇå¡ŞåĚűèćńèő¿åőŽäÿ

```
dist_strategy = fleet.DistributedStrategy()
dist_strategy.a_sync = False

optimizer = fluid.optimizer.SGD(learning_rate=0.0001)
optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)
```

8.1.5 åijĂåğŃèő■çżČ

åőŇæĹŘæĺąåđŃåŔŁèő■çżČç■ŰçŢëäžěåŘŐïijŇæĹŚäžňåřšåŕŕäžěåijĂåğŃèő■çżČæĺąåđŃäžĘãĂĆåŻääÿ

åŕźäžŐæIJ■åŁąåŹÍèŁÇçĆźïijŇéęŰåĚĹçŤÍinit_server()æŐěåŔčåŕźåĚűè£ŻèąŇåĹİåğŃåŇŰïijŇçĎ

åŘŇæăůåŕźäžŐèő■çżČèŁÇçĆźïijŇçŤÍinit_worker()æŐěåŔčè£ŻèąŇåĹİåğŃåŇŰåŘŐïijŇ
åijĂåğŃæL'ğèąŇèő■çżČäżżåŁąãĂĆè£ŘèąŇX.Trainer.fitæŐěåŔčåijĂåğŃèő■çżČïijŇåźűå¿ŮåĹřèő■çż

```
if fleet.is_server():
    fleet.init_server()
    fleet.run_server()
else:
    fleet.init_worker()
    trainer = X.Trainer(fluid.CPUPlace())
    trainer.fit(model, loader, epoch=10)
```

8.1.6 è£ŘèąŇèő■çżČèĎŽæIJň

åőŽäźL'åőŇèő■çżČèĎŽæIJňåŘŐïijŇæĹŚäžňåřšåŔŕäžěçŤÍfleetrunæŇĞäžd'è£ŘèąŇåĹĘåÿČåijŔäżżå
worker_numåĹĘåĹńäÿžæIJ■åŁąèŁÇçĆźåŠŇèő■çżČèŁÇçĆźçŽĎæŢřéĞŔãĂĆåIJÍæIJňä¿Ńäÿ■ïijŇæIJ■åŁąè

```
fleetrun --server_num=1 --worker_num=2 ctr_app.py
```

CHAPTER 9

ä¡£çŤÍInMemoryDataset/QueueDatasetè£Žèą Ňèő■çżČ

9.1 æşÍæĎŔ

æIJňæŢŹçÍŃçŻőåĽ■äÿ■æŤŕæŇĄåŁÍæĂĄåŻ¿ïijŇäžĚæŤŕæŇĄåIJÍpaddleéİŹæĂĄåŻ¿æÍąåijŔäÿŃä¡£çŤÍ

```
paddle.enable_static()
```

9.2 çőĂäžŃ

äÿžäžĘèČ¡énŸéĂ§è£ŘèąŇæÍąåđŃçŽĎèő■çżČïijŇæĹŚäžňä¡£çŤÍInMemoryDataset/QueueDatasetAPIè£ŽèąŇénŸæĂğèČ¡çŽĎIOïijŇåĚůä¡ŞäžŃçż■åŕäžěåŔĆèĂČæŰĞæąčInMemoryDat
åŠŇ QueueDataset, äžěäÿŃçőĂçğřDatasetãĂĆDatasetæŸŕäÿžåď Žçž£çÍŃåŔŁåĚÍåijĆæ■ěæŰźåijŔéĞŔèžr
åď ŽæűĹèť źèĂĚçŽĎæÍąåijŔïijŇäijŽæđĄåď ğçŽĎåŁäéĂ§æĹŚäžňçŽĎæÍąåđŃèő■çżČãĂĆ

æIJňæŰĞäžěèő■çżČword2vecæÍąåđŃäÿžä¿ŃïijŇåIJÍèő■çżČä¥■åijŢåĚěå§žäžŐDataset APIèŕžåŔŰèő■çżČæŢŕæ■őçŽĎæŰźåijŔïijŇæĹŚäžňçŽť æŐěåŁăè¡¡FleetxéćĎåĚĹåőŽäźL'åě¡çŽĎword2vecto

9.2.1 åijŢåĚědataset

1. éĂŽè£Ğdataset = paddle.distributed.InMemoryDataset() æĹŰèĂĚ dataset = paddle.distributed.QueueDataset()åĹŽåžžäÿĂäÿłDatasetåŕźèšą
2. æŇĞåőŽdatasetèŕžåŔŰçŽĎèő■çżČæŰĞäžűçŽĎåĹŮèąÍïijŇ éĂŽè£Ğset_filelistéĚ■ç¡őãĂĆ
3. éĂŽè£Ğdataset.init() api è£ŽèąŇDatasetçŽĎåĹİåğŃåŇŰéĚ■ç¡őïijŇinit()æŐěåŘčæŐěæŤű èŕęèğĄapiæŰĞæąčïijŇåĹŮäÿ¿åĞääÿłéĚ■ç¡őçŽĎåĹİåğŃåŇŰ

a. åřĘæĹŚäžňåőŽäźL'åě¡çŽĎæŢřæ■őè¿ŞåĚěæäijåijŔäijăçžŹDataset, éĂŽè£Ğuse_varéĚ■ç¡õãĆ

b. æŇĞåőŽæĹŚäžňçŽĎæŢřæ■őèŕžåŔŰæŰźåijŔïijŇçŤśmy_data_generator.pyåőđçŐřæŢřæ■őèŕžåŔŰçŽĎèğĎåĹŹïijŇåŘŐéİćářĘäijŽäžŇçż■èŕžåŔŰèğĎåĹŹçŽĎåőđçŐř, éĂŽè£ĞpipeéĚ■ç¡õãĆpipe_commandæŸŕDatasetçL'źæIJL'çŽĎéĂŽè£Ğçőąé

c. æŇĞåőŽæŢřæ■őèŕžåŔŰçŽĎbatch_sizeïijŇéĂŽè£ĞbatchéĚ■ç¡õãĆ

d. æŇĞåőŽæŢřæ■őèŕžåŔŰçŽĎçž£çĺŇæŢřïijŇäÿÄèĹňèŕěçž£çĺŇæŢřåŠŇèő■çżČçž£çĺŇåžŤä£İæŇĄ

```
dataset = paddle.distributed.InMemoryDataset()
batch_size = config.config["batch_size"]
thread_num = config.config["thread_num"]
dataset.init(use_var=model.inputs, pipe_command="python my_data_
↪generator.py", batch_size=batch_size, thread_num=thread_num)
dataset.set_filelist([config.config["train_files_path"]])
```

9.2.2 åęĆä¡ŢæŇĞåőŽæŢřæ■őèŕžåŔŰèğĎåĹŹ

åIJĺäÿŁæŰĞæĹŚäžňæŔŘåĽřäžĘçŤśmy_data_generator.pyåőđçŐřåĚůä¡ŞçŽĎæŢřæ■őçőąéĄŞèŕžåŔŰèğĎåĹŹïijŇéĆčäźĹïijŇæĂÓæăůäÿždatasetåĹŽåžžæŢřæ■őèŕžåŔäžěäÿŇæŸŕmy_data_generator.pyçŽĎåĚíéČĺäžčçăĄïijŇåĚůä¡ŞæţĄçĺŇåęĆäÿŇïijŽ

1. éęŰåĚĹæĹŚäžňéIJĂèęĄåijŢåĚědata_generatorçŽĎçśžïijŇä¡■äžŐpaddle.distributed.fleet.data_generatorãĂĆ 2. åčřæŸŐäÿĂäžŽåIJĺæŢřæ■őèŕžåŔŰäÿ■äijŽçŤĺåĽřçŽĎ
3. åĹŽåžžäÿĂäÿłå■ŘçśżWord2VecReaderïijŇçžğæL'£fleet.data_generatorçŽĎå§žçśžïijŇå§žçśžæIJL'åď'Žçğ■éĂL'æŃľïijŇåęĆæđIJæŸŕåď'Žçğ■æŢřæ■őçśžåđŃæů
4. çžğæL'£åźűåőđçŐřå§žçśžäÿ■çŽĎgenerate_sampleåĞ¡æŢřïijŇéĂŘèąŇèŕžåŔŰæŢřæ■őãĆčèŕěåĞ¡æŢ
5. åIJĺè£Źäÿłåŕŕäžěè£■äžčçŽĎåĞ¡æŢřäÿ■ïijŇåęĆçď'žä¿ŃäžčçăĄäÿ■çŽĎdef nce_reader()ïijŇæĹŚäžňåőŽäźL'æŢřæ■őèŕžåŔŰçŽĎéĂžè¿ŚãĂĆä¿ŃåęĆåŕźäžěèąŇäÿžå■Ţä¡■çŽĎæŢřæ

6. æIJĂåŘŐïijŇæĹŚäžňéIJĂèęĄåřĘæŢřæ■őæŢt'çŘĘäÿžçL'źåőŽçŽĎbatchçŽĎæăijåijŔïijŇæL'■èČ¡åď'§èć æăźæ■őèő¿åőŽçŽĎâĂŹbatch_sizeâĂŹ, èŕěåĞ¡æŢřäijŽåIJĺgenerator_sampleåĞ¡æŢřäžğçŢ§æăůæ local_iter()ãĂĆ

7. çőĂå■ŢæİèèŕťïijŇæŢřæ■őçŽĎè¿ŞåĞžéąžåžŔäÿŐæĹŚäžňåIJĺç¡ŚçžIJäÿ■åĹŽåžžçŽĎinputså£Ěéąžæ [value]),('true_label',[value]),('neg_label',[value])]

```
import sys
import io
import os
import re
import collections
import time
import config
import logging

import paddle
import numpy as np
import paddle.distributed.fleet as fleet
```

(äÿŃéąţçžğçż■)

(çż■äÿŁéąţ)

```
logging.basicConfig(format='%(asctime)s - %(levelname)s -
↪%(message)s')
logger = logging.getLogger("fluid")
logger.setLevel(logging.INFO)

class NumpyRandomInt(object):
    def __init__(self, a, b, buf_size=1000):
        self.idx = 0
        self.buffer = np.random.random_integers(a, b, buf_size)
        self.a = a
        self.b = b

    def __call__(self):
        if self.idx == len(self.buffer):
            self.buffer = np.random.random_integers(self.a, self.b,
                                                    len(self.
↪buffer))
            self.idx = 0

        result = self.buffer[self.idx]
        self.idx += 1
        return result

class Word2VecReader(fleet.MultiSlotDataGenerator):
    def init(self,
             dict_path,
             nce_num,
             window_size=5):

        self.window_size_ = window_size
        self.nce_num = nce_num

        word_all_count = 0
        id_counts = []
        word_id = 0

        with io.open(dict_path, 'r', encoding='utf-8') as f:
            for line in f:
                word, count = line.split()[0], int(line.split()[1])
                word_id += 1
                id_counts.append(count)
                word_all_count += count

        self.word_all_count = word_all_count
        self.corpus_size_ = word_all_count
        self.dict_size = len(id_counts)
        self.id_counts_ = id_counts
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
        logger.info("corpus_size:", self.corpus_size_)
        self.id_frequencys = [
            float(count) / word_all_count for count in self.id_
↪counts_
        ]
        logger.info("dict_size = " + str(self.dict_size) + " word_
↪all_count = " + str(word_all_count))

        self.random_generator = NumpyRandomInt(1, self.window_size_␣
↪+ 1)

    def get_context_words(self, words, idx):
        """
        Get the context word list of target word.
        words: the words of the current line
        idx: input word index
        window_size: window size
        """
        target_window = self.random_generator()
        start_point = idx - target_window  # if (idx - target_
↪window) > 0 else 0
        if start_point < 0:
            start_point = 0
        end_point = idx + target_window
        targets = words[start_point:idx] + words[idx + 1:end_point␣
↪+ 1]
        return targets

    def generate_batch(self, samples):
        def local_iter():
            np_power = np.power(np.array(self.id_frequencys), 0.75)
            id_frequencys_pow = np_power / np_power.sum()
            cs = np.array(id_frequencys_pow).cumsum()
            result = [[], []]
            for sample in samples:
                tensor_result = [("input_word", []), ("true_label",␣
↪[]), ("neg_label", [])]
                tensor_result[0][1].extend(sample[0])
                tensor_result[1][1].extend(sample[1])
                neg_array = cs.searchsorted(np.random.sample(self.
↪nce_num))

                tensor_result[2][1].extend(neg_array)

                yield tensor_result
        return local_iter
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
    def generate_sample(self, line):
        def nce_reader():

            word_ids = [int(w) for w in line.split()]
            for idx, target_id in enumerate(word_ids):
                context_word_ids = self.get_context_words(
                    word_ids, idx)
                for context_id in context_word_ids:
                    yield [target_id], [context_id]
        return nce_reader

if __name__ == "__main__":
    my_data_generator = Word2VecReader()
    my_data_generator.init(config.config["dict_path"], config.
↪config["nce_num"])
    my_data_generator.set_batch(config.config["batch_size"])

    my_data_generator.run_from_stdin()
```

9.2.3 å£ńéĂ§èřČèŕŢDataset

æĹŚäžňåŕŕäžěèĎśçężçžĎç¡ŚæđűæđĎïijŇå■ŢçŃňéłŇèŕĄDatasetçŽĎè¿ŞåĞžæŸŕåŘęçňęåŘĹæĹŚäžňécĎ
cat æŢřæ■őæŰĞäžű | python datasetèŕżåŘŰpythonæŰĞäžűè£ŻèąŇdatasetäžčçăĄçŽĎèřČèŕŢ

```
cat train_data/part_912 | python my_data_generator.py
```

è¿ŞåĞžçŽĎæŢřæ■őæăijåijŔåęĆäÿŃïijŽ input_word:size ; input_word:value ; true_label:size ; true_label:value ; neg_label:size ; neg_label:value

çŘĘæČşçŽĎè¿ŞåĞžäÿž(æĹłåŔŰäžĘäÿĂäÿłçL'Ğæőţ)ïijŽ

```
...
1 112 1 2739 5 6740 451 778 90446 3698
...
```

ä¡£çŤĺDatasetçŽĎäÿĂäžŻæşĺæĎŔäžŃéąź - DatasetçŽĎå§žæIJňåŐ§çŘĘïijŽåřĘæŢřæ■őprintåĽřçijŞå■Ÿ
- datasetçŻőåL'■åŔłæŤŕæŇĄåIJĺunbuntuåŔŁCentOSç■L'æăĞåĞĘLinuxçŐŕåćČäÿŃä¡£çŤĺïijŇåIJĺw

9.2.4 æŢřæ■őåĞĘåď'Ğ

åŕŕäžěåŔĆèĂČæŰĞæąč çŽĎæŢřæ■őåĞĘåď'ĞéČĺåĹĘ åőŇæŢt' æŢřæ■őäÿŃè¡¡äžěåŔŁécĎåď'ĎçŘĘäźŃ

```
mkdir demo_train_data
cp train_data/part_1 demo_train_data/
```

9.3 èő■çżČ

æĹŚäžňæŁŁåŐ§æİěçŽĎèő■çżČäžčçăĄ:

```
trainer = X.CPUTrainer()
trainer.fit(model, loader, epoch=10)
```

æŻ£æ■ćæĹŘåęĆäÿŃä¡£çŤĺDatasetèő■çżČçŽĎætĄçĺŃ, æĹŚäžňäžěäÿĂäÿłepochäÿžä¿ŃïijŽ

```
import paddle
import paddle.fluid as fluid
import paddle.distributed.fleet as fleet
import config
# åijĂåŘŕpaddleéİŹæĂĄåŻ¿æĺąåijŔ
paddle.enable_static()

fleet.init()

model = X.applications.Word2vec()

"""
need config loader correctly.
"""

loader = model.load_dataset_from_file(train_files_path=[config.
↪config["train_files_path"]], dict_path=config.config["dict_path"])

dist_strategy = fleet.DistributedStrategy()
dist_strategy.a_sync = True

optimizer = fluid.optimizer.SGD(learning_rate=0.0001)
optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)

if fleet.is_server():
    fleet.init_server()
    fleet.run_server()
else:
    place = paddle.CPUPlace()
    fleet.init_worker()
    exe = paddle.static.Executor(place)
    default_startup_program = paddle.static.Program()
    default_main_program = paddle.static.Program()
    scope1 = fluid.Scope()
    with fluid.scope_guard(scope1):
        exe.run(model.startup_prog)

    dataset = paddle.distributed.QueueDataset()
```

(äÿNéąţçżğçż■)

(çż■äÿŁéąţ)

```
    batch_size = config.config["batch_size"]
    thread_num = config.config["thread_num"]
    dataset.init(use_var=model.inputs, pipe_command="python my_data_
↪generator.py", batch_size=batch_size, thread_num=thread_num)
    dataset.set_filelist([config.config["train_files_path"]])

    with fluid.scope_guard(scope1):
        exe.train_from_dataset(model.main_prog, dataset, scope1,␣
↪debug=False, fetch_list=[model.loss], fetch_info=["loss"], print_
↪period=10)

    fleet.stop_worker()
```

æIJĂåŘŐæůżåŁääÿŁè£řäžčçăĄä¡£çŤÍçŽĎéĚ■ç¡őæŰĞäžűconfig.py

```
config = dict()

config["dict_path"] = "thirdparty/test_build_dict"
config["train_files_path"] = "demo_train_data/part_1"
config["batch_size"] = 1000
config["nce_num"] = 5
config["thread_num"] = 12
```

éĂŽè£Ğäžěäÿ Łçő Äæť ĄçŽĎäžčçăĄïijŇå■şåŔŕäžěåőđçŐřword2vectoræĺąåđŃçŽĎåď Žçž£çÍŃåźűåŔŚèő

CHAPTER 10

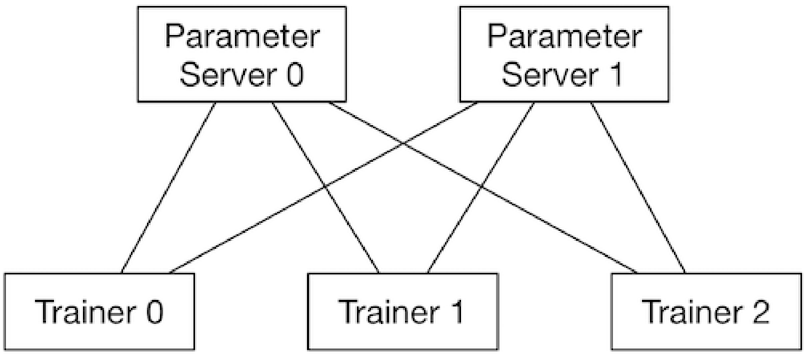
Collective åŘŇæ■ěèő■çżČåőđèůţ

10.1 åŘŇæ■ěèő■çżČçőĂäżŃ

èőÿåď'ŽçăŤçľ'űèąĺæŸŐæůśåžęå■ęäźăçŽĎéćĎèő■çżČåŔŮçŽŁäžŐæŻť'åď'ŽçŽĎæŢřæ■ő[1] [2] [3]ïijŇä¡ĘæŻť'åď'ğçŽĎæŢřæ■őéĞŔäź§æĎŔåŞşçİĂæŻť'éŢ£çŽĎèő■çżČèĂŮæŮűïijŇæŢřæ■őåźűèąŇåŘŇ

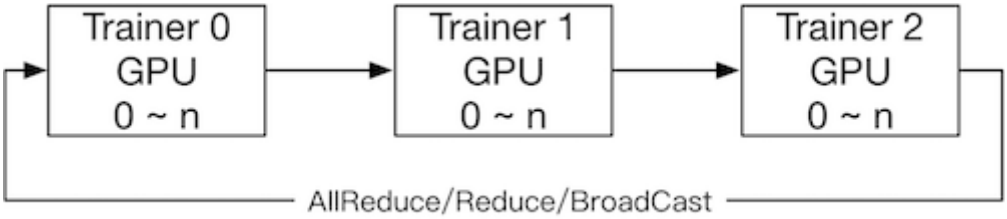
åŘŇæ■ěèő■çżČéĂŽè£ĞæŢřæ■őåĹŠåĹĘïijŇåřĘèőąçőŮåůĕä¡IJéĞŔïijĹåĽ■åŘŠãĂĄåŔ■åŘŠïijĽåĹĘåÿČ éŻĘç¿ď'äÿ■çŽĎæŕŔäÿĂäÿłworkeräÿĿïijŇ æŔŘéńŸæŢť'ä¡Şèőąçő'ŮåŘđåŘŘãĂĆä¡ĘåŔĆæŢřæŻť'æŰř(update) çŽĎè£ĞçĺŃåĲĺäÿd'çğ■æĺąåijŔäÿ■æĲĽæĽ'Ăäÿ■åŘŇïijŽ

- åĲĺPServeræĺąåijŔäÿ■ïijŇäijŽåŘŕåŁĺåď'Žäÿłpservers åŠŇåď'Žäÿłtrainer-sïijŇæŕŔäÿłpserveräijŽä£İå■ŸäÿĂéČĺåĹĘæĺąåđŃåŔĆæŢřïijŇåźűèť'§èť'čæŐěæŤűäżŐtraineråŔŠéĂĄçŽĎ äijŽä£İå■ŸäÿĂäż¡åőŇæŢť'çŽĎæĺąåđŃïijŇåźűä¡£çŤĺäÿĂéČĺåĹĘæŢřæ■őè£Żèąňèő■çżČïijŇçĎűåŘŐåŘ pserverè£ŽçĺŃåŠŇtraineråŔŕäżěåĲĺäÿ■åŘŇçŽĎèőąçőŮèŁĆçĆźäÿĿïijŇäź§åŔŕäżěåĲĺåŘŇäÿĂåĚňçŤĺè



- åĲĺCollectiveæĺąåijŔäÿ■ïijŇéŻĘç¿ď'äÿ■åŘłå■ŸåĲĺåď'ŽäÿłåĲŕä¡■åźşç■Ľ'çŽĎtrainersãĂĆ æŕŔäÿłtrainerè£ŽçĺŃéČ¡ä£İå■ŸäÿĂäż¡åőŇæŢť'çŽĎæĺąåđŃåŔĆæŢřãĂĆ åĽ■åŘŠåŠŇåŔ■åŘŠäÿ■æŕŔäÿł trainer ä¡£çŤĺèĞłåůśåĹŠåĹĘ (shardïijĽ'çŽĎæŢřæ■őè£ŻèąŇèőąçőŮïijŇå¿ŮåĹřåŕźåžŤçŽĎæćŕåžęïijŻäźŃåŘŐtrainers äźŃéŮť'éĂŽè£Ğ allreduce ç■Ľ' Collective éĂŽä£ąæŰźåijŔ[4]

åŘŇæ■ěæćŕåžęåĹřæĽĂæIJĽtrainersïijŇæIJĂåŘŐæŕŔäÿł trainer ä¡£çŤĺåŘŇæ■ěåŘŐçŽĎæćŕåžęçŃňçńŇåőŇæĹŘåŔĆæŢřæŻťæŰřãĂĆ



çŻÿäžď äžŐåijĆæ■ěèő■çżČ, åŘŇæ■èèő■çżČçŽĎçŽĎäijŸåŁ£åIJĺäžŐLossåŔŕäžěæŕŤè¿ČçĺşåőŽçŽĎäÿŃ åŻăæ■ď åIJĺèő■çżČè¿Čäÿžåď■æİĆçŽĎæĺąåđŇæŮűïijŇå■şæĺąåđŃèő■çżČè£ĞçĺŃäÿ■çěđçżŔç¡ŚçżIJèő■çżČè

Fleetäÿ■ PServeræĺąåijŔä¡£çŤĺ gRPC éĂŽä£ąïijŇCollectiveæĺąåijŔä¡£çŤĺ NCCL2 éĂŽä£ąãĂĆ

äÿŃæŰĞåřEçŤśäÿĽéČĺåĹEçżĎæĹŘïijŽ

- äžŃçż■ Fleet åŘŇæ■èèő■çżČäÿ■åÿÿçŤĺçŽĎåĞääÿłç■ŰçŢě ãĂĄäijŸåŇŰ
- çżŞåŘĹäÿŁè£řåÿÿçŤĺäijŸåŇŰïijŇçżŹåĞžäÿĂäÿłåIJĺ 4èŁĆçĆź 32 V100 éŻEç¿ď èő■çżČ ResNet50çŽĎçď žä¿Ńäžčçă Ą
- åőŇæŢť Fleet åŘŇæ■èèő■çżČåŔĆæŢřç■ŰçŢěäżŃçż■

10.2 Fleet Collective åŘŇæ■èèő■çżČäijŸåŇŰ

Fleet æŤŕæŇĄåIJĺ GPU (CUDA çĽĹæIJň >= 7.5) æIJ■åŁąåŹĺéŻEç¿ď äÿŁåőŇæĹŘéńŸæĂğèČ¡åĹEåÿČåijŔèő■çżČãĂĆ çŤĺæĹůåŔŕäžěéĂŽè£Ğ `fleet.DistributedStrategy` èő¿ç¡őèöÿåď ŽäÿŐèő■çżČæĂğèČ¡ç■ŰçŢěçŻÿåĚşåŔĆæŢřãĂĆçŽőåĽ■Fleet äÿžè£ŹäžŻåŔĆæŢřæŔŘä¿ŻäžEäÿĂäÿłè¿ČéĂŽçŤĺéżŸèődď åĂijïijŇçŤĺæĹůåŔŕäžěäÿ■åŐżèřČæŢť ãĂĆä¡EåęĆ

åIJĺè£ŻèąŇæĂğèČ¡äijŸåŇŰæŮűïijŇ æčĂæ§ěæŕŔéąźäijŸåŇŰçĆźåźűéłŇèŕĄåŕźåžŤæŔŘå■ĞïijŇæIJĂçż äÿĂäÿłçőĂå■ŢçŽĎéłŇèŕĄå¡ŞåĽ■çŽĎèő■çżČçĺŃåžŔæŸŕåŘęéIJĂèęĄè£ŻäÿĂæ■ěäijŸåŇŰæĂğèČ¡çŽĎæŰź æŸŕæ§ěçIJŃGPUçŽĎèőąçőŮåĹl'çŤĺçŐĞïijŇéĂŽåÿÿçŤĺ *nvidia-smi* åŚ¡äžď æ§ěçIJŃãĂĆ åęĆæđIJGPUåĹl'çŤĺçŐĞè¿Čä¡ŐïijŇåĹŹåŔŕèČ¡å■ŸåIJĺè¿Čåď ğçŽĎäijŸåŇŰçl'žéŮt'ãĂĆ

äÿŃæŰĞåřĘäżŃçż■åŕźæĂğèČ¡å¡śåŞ■è¿Čåď ğïijŇèő¿ç¡őéćŚçŐĞæŕŤè¿ČéńŸçŽĎåĞääÿłåŔĆæŢřïijŇèŕę

æşĺæĎŔïijŽ ä¡£çŤĺNCCL2æĺąåijŔåĹĘåÿČåijŔèő■çżČæŮűïijŇéIJĂèęĄçąőä£İæŕŔäÿłèŁĆçĆźèő■çżČç■L

- éŽŔæIJžéĞĞæăůäÿĂäžŻæŢřæ■őïijŇèąěåĚĺåĹĘéĚ■åĹřè¿ČåřŚæŢřæ■őçŽĎèŁĆçĆźäÿŁãĂĆïijĹæŐĺè■Ř
- åIJĺpythonäžčçăĄäÿ■ïijŇærŔäÿłèŁĆçĆźærŔäÿłpassåŘłèő■çżČåŻžåőŽçŽĎbatchæŢřïijŇåęĆæđIJè£Źäÿ

10.2.1 OPèđ■åŘĹ

åřĘæĺąåđŃç¡ŚçżIJäÿ■éąžåžŔæĽğèąŇçŽĎåď ŽäÿłOPsè£ŻèąŇèđ■åŘĹèČ¡åď §åĞŔåřŚOP èřČåžęçŽĎåijĂéŤĂïijŇæŔŘå■Ğèő■çżČéĂ§åžęãĂĆçŽőåĽ■Fleet äÿ■æŤŕæŇĄåęĆäÿŃ3çğ■çŽĎOP èđ■åŘĹïijŽ

- `fuse_all_optimizer_ops`ïijŽèąÍæŸŐæŸŕåŘęèđ■åŘĹ(fuse) æŸŕåŘęèđ■åŘĹ optimizer_opïijŇäżĚåŕźéČĺåĹĘ optimizer åŘŕçŤĺïijĹSGDãĂĄAdamåŠŇMomentumïijL'ãĂĆ
- `fuse_elewise_add_act_ops`ïijŽèąÍæŸŐæŸŕåŘęèđ■åŘĹ(fuse) elementwise_add_opåŠŇactivation_opãĂĆ
- `fuse_bn_act_ops`ïijŽèąÍæŸŐæŸŕåŘęèđ■åŘĹ(fuse) batch_norm_op åŠŇ activation_opãĂĆ

éĂŽåÿÿä¡£çŤĺè£ŹäžŽç■ŰçŢěéČ¡äijŽä¡£æŢt'ä¡ŞæL'ğèąŇè£ĞçÍŃæŻt'å£ńãĂĆ

```
dist_strategy = fleet.DistributedStrategy()
dist_strategy.fuse_all_optimizer_ops = True
dist_strategy.fuse_bn_act_ops = True
dist_strategy.fuse_elewise_add_act_ops = True
```

10.2.2 AllReduceèđ■åŘĹ

AllReduce èđ■åŘĹézŸèŏd'æČĚåĘţäÿŃäijŽårĘåŘŇäÿĂlayeräÿ■åŔĆæŢřçŽĎæćŕåžęçŽĎåd'ŽäÿłAllReduceæŞ■ä¡IJåŘĹåźűæĹŘäÿĂäÿłãĂĆ
æŕŤåęĆåŕźäžŐ fluid.layers.fc äÿ■æIJL'WeightåŠŇBiasäÿd'äÿłåŔĆæŢřïijŇæL'ŞåijÄèřéĂL'éąźäźŃåL'■ïijŇéIJ
æŞ■ä¡IJãĂĆè£ŹæăůåŔŕäžěåĞŔåřŠæćŕåžęåŘŇæ■ěæŮűçŽĎéĂŽä£ąèĂŮæŮűãĂĆ

æ■ď'åd'Űïij Ňäÿžæ Ťŕæ ŇĄæŻt'åd'ğçšŠåžęçŽĎåŔĆæŢřæćŕåžęèđ■åŘĹïijŇFleet
æŔŘä¿ŻäžĘäžěäÿŃäÿd'äÿłéĂL'éąźïijŇçŤĺæĹůåŔŕäžěåIJĺèő■çżČçÍŃåžŔè£ŘèąŇåL'■åIJÍDistributedStrategyäÿ

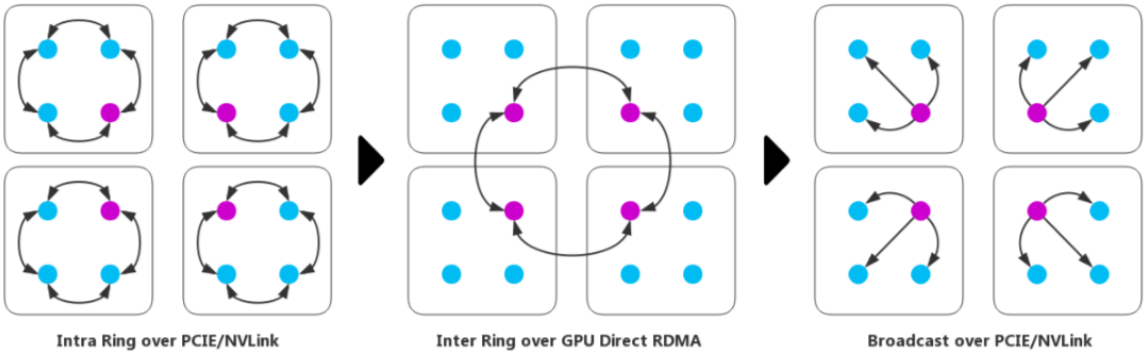
- `fuse_grad_size_in_MB`: æŇĞåőŽæŕŔäÿłAllReduceæŞ■ä¡IJçŽĎæćŕåžęå■ŮèŁĆæŢřïijŇåęĆèřéåŔ åĹŹæŕŔæňąAllReduceèřČçŤĺäijäè¿Ş16MBçŽĎæćŕåžęĂĆ èřéåŔĆæŢřçŽĎçżŔéłŇåĂijäÿžæĂżéĂŽä£ąéĞŔçŽĎå■ĄåĹĘäźŃäÿĂãĂĆ
- `fuse_grad_size_in_TFLOPS`: æŇĞåőŽæŕŔæňąAllReduceæŞ■ä¡IJçŽĎæIJĂåd'ğåśĆæŢřïijŇå■şåĹřè¿¿èŕéåśĆæŢřåŕśè£ŻèąŇAllReduceãĂĆåęĆèřéåŔĆæŢřç■ åĹŹæIJĂåd'ŽæŕŔ50åśĆåĄŽäÿĂæňą fused AllReduceãĂĆ

æşĺæĎŔïijŽ AllReduceèđ■åŘĹçŻőåL'■äÿ■æŤræŇĄsparseåŔĆæŢřæćŕåžęãĂĆ

```
dist_strategy = fleet.DistributedStrategy()
dist_strategy.fuse_grad_size_in_MB=16
dist_strategy.fuse_grad_size_in_TFLOPS=50
dist_strategy.fuse_all_reduce_ops=True
```

10.2.3 åĹĘåśĆ AllReduce

åŕźäžŐåd'ŽæIJžæĺąåijŔïijŇéŠĹåŕźårŔæŢřæ■őéĞŔçŽĎéĂŽä£ąïijŇRing AllReduceéĂŽä£ąæŢĹçŐĞä¡ŐïijŇéĞĞçŤĺHierarchical AllReduceåŔŕäžěçijŞèğčè£ŹäÿĂéŮőécŸãĂĆ åĹĘåśĆAllReduce è£ŘèąŇåęĆäÿŃåŻ¿æL'Äçd'žïijŽ




```
dist_strategy = fleet.DistributedStrategy()
dist_strategy.use_hierarchical_allreduce = True
dist_strategy.hierarchical_allreduce_inter_nranks = 8
```

10.2.4 ä¡£çŤÍåŘŇæ■ěAllreduce

Fleet ä¡£çŤÍåď Žè£ŻçÍŃ+NCCL2æÍąåijŔïijÍcollectiveïijL'äžèèŐůå¿ŮæŻť'åě¡çŽĎæĂğèČ¡ãĂĆ åIJÍåď Žè£ŻçÍŃæÍąåijŔäÿŃïijŇæŕŔåŘřæIJ■åŁąåŹÍçŽĎæŕŔäÿłGPUå■ąéČ¡äijŽåŕźåžŤåŘŕåŁÍäÿĂäÿłèő■çżČè£ éŻEç¿d'äÿ■çŽĎæL'ĂæIJL'è£ŻçÍŃäźŃéŮť'äijŽäžŠçŻÿéĂŽä£ąåőŇæŁŘèő■çżČãĂĆäžěæ■d'æŰźåijŔæIJĂåď ğ

```
dist_strategy.sync_nccl_allreduce=True
```

10.2.5 èő¿ç¡őåŘĹéĂĆçŽĎnccléĂŽä£ąåŹÍæŢřéĞŔ

nccléĂŽä£ąåŹÍæŢřéĞŔ nccl_comm_num åŔŕäžěåŁåå£ńGPUäźŃéŮť'çŽĎéĂŽä£ąæŢĹçŐĞïijŇåžžèőőå■

```
dist_strategy = fleet.DistributedStrategy()
dist_strategy.nccl_comm_num = 2
```

10.2.6 èő¿ç¡őåŘĹéĂĆçŽĎCPUçž£çÍŃæŢř

PaddlePaddle Fluidä¡£çŤÍâĂIJçž£çÍŃæśăâĂİ [5] æÍąåđŃèřČåžęåźűæL'ğèąŇOpïijŇOpåIJÍåŘŕåŁÍG-PUèőąçőŮäźŃåL'■ïijŇ éĂŽåÿÿéIJĂèęĄCPUçŽĎå■ŔåŁl'ïijŇçĎűèĂŇåęĆæđIJOpæIJňèžńå■ăçŤÍæŮűéŮť'å¿Ĺ

æăźæ■őäžěå¿ĂçŽĎçżŔéłŇïijŇåŕźäžŐCPUäżžåŁąïijŇnum_threads=2 * dev_count æŮűæĂğèČ¡è¿Čåě¡ïijŇåŕźäžŐGPUäżžåŁąïijŇnum_threads=4 * dev_count æŮűæĂğèČ¡è¿Čåě¡ãĂĆæşÍæĎŔïijŽçž£çÍŃæśääÿ■æŸŕèűŁåď ğèűŁåě¡ãĂĆ

```
dist_strategy = fleet.DistributedStrategy()
dist_strategy.thread_num = 3
```

10.2.7 æŔŘéńŸç¡ŚçżIJçŽĎåŘđåŘŘ

åď ŽèŁĆçĆźèő■çżČæŮűç¡ŚçżIJçŽĎåÿęåő¡åÿÿåÿÿæĹŘäÿžèő■çżČçŽĎçŞűéĆĹãĂĆæĹŚäžňåIJĺåőđæţŃäÿ
socket çŽĎéĂŽä£ąæŰźåijŔåřĘæĹŘäÿžèő■çżČéĂ§åžęçŽĎçŞűéĆĹïijŇ ä¡£åď ŽèŁĆçĆźèő■çżČæŮăæşŢåĚĚåĹĘåĹľçŤĺ FLeet æůůåŘĹçš¿åžęèőąçőŮåÿęæİěçŽĎéĂ§åžęæŔŘå■
åIJĺæĹŚäžňåőđæţŃäÿ■ä¡£çŤĺ: 100Gb ç¡Śå■ąïijŇRDMA[7] åŠŇ InfiniBand[8]æİěæŔŘå■Ğç¡ŚçżIJåÿęåő¡ïijŇä¡£ç¡ŚçżIJäijäè¿Şäÿ■äijŽæĹŘäÿžèőąçőŮéĂ§åžęçŽĎçŞűéĆĹ åIJĺåijĂåğŃèő■çżČåĽ■ïijŇéIJĂèęĄæ■čçąőèő¿ç¡őäžěäÿŃ NCCL çŐŕåćČåŔŸéĞŔä¡£åŕźåžŤçąňäžűèő¿ç¡őçŤ§æŢĹïijŽ

Env Name	Description
NCCL_SOCKET_IFNAME	The RDMA device, e.g. eth2
NCCL_P2P_DISABLE	Set to 1 to disable P2P transfer between GPUs
NCCL_IB_DISABLE	Set to 1 to disable using RDMA
NCCL_IB_CUDA_SUPPORT	Set to 1 to enable GPU Direct if supported
NCCL_DEBUG	Set debug level: VERSION, WARN, INFO

10.2.8 éćĎåĚĹåĹĘéĚ■èűşåď§çŽĎæŸ¿å■Ÿ

éĂŽè£ĞçŐŕåćČåŔŸéĞŔ FLAGS_fraction_of_gpu_memory_to_use=0.7 èő¿ç¡őéćĎåĚĹåĹĘéĚ■çŽĎæŸ¿å■Ÿå■ăæŕŤãĂĆ çŤśäžŐCUD-AåŐ§çŤ§çŽĎæŸ¿å■ŸåĹĘéĚ■cuMallocåŠŇéĞŁæŤ¿cuFreeæŞ■ä¡IJåİĞæŸŕåŘŇæ■ěæŞ■ä¡IJïijŇéİđåÿÿèĂŮæ éĂŽè£Ğ èő¿ç¡ő FLAGS_fraction_of_gpu_memory_to_use æĹŘäÿĂäÿłè¿Čåď ğçŽĎåĂijïijŇæŕŤåęĆ0.7ïijŇåŔ

0.7 æŸŕæŇĞ 70%çŽĎæŸ¿å■ŸäijŽéćĎåĚĹåĹĘéĚ■ãĂĆèő¿ç¡őçŽĎèŇČåŻť æŸŕ0.0~1.0ãĂĆ

```
os.environ['FLAGS_fraction_of_gpu_memory_to_use'] = "0.98"
```

10.2.9 éŹ■ä¡Őscope dropéćŚçŐĞåŠŇfetchéćŚçŐĞ

åĞŔåřŚscope dropåŠŇfetchéćŚçŐĞïijŇåŔŕäžěåĞŔåřŚéćŚçźĄçŽĎåŔŸéĞŔåĘĚå■ŸçŤşèŕůãĂĄéĞŁæŤ¿äžŐèĂŇæŔŘå■ĞæĂğèČ¡ãĂĆ

```
# æŕŔ 30 batch äźŃåŘŐæÿĚçŘĘäÿĂæňąäÿt'æŮűåŔŸéĞŔ
dist_strategy = fleet.DistributedStrategy()
dist_strategy.BuildStrategy = {'num_iteration_per_drop_scope': 30}

# éŹ■ä¡ŐfetchéćŚçŐĞïijŇæŕŔ 30 batch fetch äÿĂæňąèő■çżČè¿ŞåĞž
for pass_id in xrange(PASS_NUM):
    batch_id = 0
    while True:
        if batch_id % 30 == 0:
            fetched = exe.run(fetch_list)
        else:
            exe.run([])
```

10.2.10 åćđåd'ğbatch_size

åĹĘåÿČåijŔåŘŇæ■ěèő■çżČïijŇèůÍèŁĆçĆźĚĂŽä£ąæĹŰåd'ŽæĹŰåřŚäijŽåÿęæİěæĂğèČ¡å¡śåŞ■ïijŇåćđåc
åŕŕäžěä£ÍæŇĄéĂŽä£ąåijĂéŤĂäÿ■åŔŸçŽĎæČĚåĘţäÿŇïijŇåćđåd'ğèőąçőŮåŘđåŘŘäżŐèĂŇéŹ■ä¡ŐéĂŽä£ąå

10.2.11 ä¡£çŤÍ DALI reader

æŢřæ■őèŕžåŔŰçŽĎäijŸåŇŰåIJÍGPUèő■çżČäÿ■èĞşåĚşéĞ■èęĄïijŇářd'åĚűåIJÍäÿ■æŰ■åćđåŁăbatch_siz
åŕŕèČ¡æĹŘäÿžèő■çżČéĂ§åžęçŽĎçŞűéćĹãĂĆ Fleet äÿ■åŕŕäžěä¡£çŤÍ Nvidia DALI6
ä¡IJäÿžæŢřæ■őloader. ä¡£çŤÍDALIçŽĎäijŸçĆźæIJL'ïijŽ

- ä¡£çŤÍGPUåőŇæĹŘéČÍåĹĘæŢřæ■őécĎåd'ĎçŘĘïijŇåŁăéĂ§æŢřæ■őèŕžåŔŰè£ĞçÍŃïijŇåĞŘářŚ CPU èt'§æŃĚãĂĆ
- DALI æŔŘä¿ŻéćĎåŔŰéŸ§åĹŮïijĹperfetch queueïijL'åŁ§èČ¡ïijŇèől'æŢřæ■őécĎåd'ĎçŘĘåŠŇæĺąåđŃè

```
import fleetx as X
model = X.applications.Resnet50()
loader = model.load_imagenet_from_file("/pathto/imagenet/train.txt",
↪ use_dali=True)
```

10.2.12 ä¡£çŤÍæůůåŘĹçš¿åžęèő■çżČ

V100 GPUæŔŘä¿ŻäžĘ Tensor Core åŕŕäžěåIJÍæůůåŘĹçš¿åžęèőąçőŮ
åIJžæŹŕæđĄåd'ğçŽĎæŔŘå■ĞæĂğèČ¡ãĂĆä¡£çŤÍæůůåŘĹçš¿åžęèőąçőŮçŽĎä¿Ńå■ŘåŔŕäžěåŔĆèĂČæŰĞæą
‘ <https://todo/>‘__

çŽőåL'■PaddleåŔłæŔŘä¿ŻåIJÍäÿd'äÿłæÍąåđŃïijĹResNet, BERTïijL'çŽĎæůůåŘĹçš¿åžęèőąçőŮåőđçŐřåź
loss scalingïijŇåĚűäžŰæÍąåđŃä¡£çŤÍæůůåŘĹçš¿åžęäź§ åŕŕäžěåŔĆèĂČäžěäÿŁçŽĎåőđçŐřåőŇæĹŘéłŇèřĄãĂ

10.3 ResNet50èő■çżČçd'žä¿Ń

èřŢéłŇåijĂågŃåL'■æĹŚäžňåůšçżŔåIJÍGPU éŻĘç¿d'äÿ■æŔŘåL'■éĚ■ç¡őåě¡ *RDMA* åŠŇ
*InfiniBand*ïijŇåĞŘářŚç¡ŚçżIJéĂŽä£ąçŽĎçŞűéćĹïijŇéĚ■ç¡őçżĘèŁĆåŠŇåĚůä¡ŞçąňäžűçŻÿåĚşïijŇåŕŕäžěåŔĆ
x] <https://community.mellanox.com/s/article/what-is-rdma-x>‘__

10.3.1 èő¿ç¡ő AllReduceèđ■åŘĹç■L'åŔĆæŢř

æćŕåžęèđ■åŘĹäÿ■çŽĎ16 åŠŇ 50 æŸŕæĹŚäžňæăźæ■őèĞłèžńç¡ŚçżIJçąňäžűåŠŇResNet50
èő■çżČèŕŢéłŇå¿ŮåĞžçŽĎçżŔéłŇåĂijïijŇçŤÍæĹůåŕŕäžěæăźæ■őèĞłèžńçąňäžűåŠŇæĺąåđŃè£ŻèąŇèřČæŢt'ã
0.7 æŸŕäÿžäžĘçżŹ DALI loader æŔŘåL'■écĎçŢŹæŸ¿å■ŸçľžéŮt'ãĂĆ

```
import os
os.environ['FLAGS_fuse_parameter_memory_size'] = "16"
os.environ['FLAGS_fuse_parameter_groups_size'] = "50"
os.environ['FLAGS_fraction_of_gpu_memory_to_use'] = "0.7"
```

10.3.2 æůżåŁää¿ìèţŰ

```
import os
import fleetx as X
import paddle
import paddle.fluid as fluid
import paddle.distributed.fleet.base.role_maker as role_maker
import time
import paddle.distributed.fleet as fleet
```

10.3.3 åőŽäźĽåĹĘåÿČåijŕæĺąåijŕåźűåĹlåğŃåŇŰæĺąåđŃåŠŇreader

è£ŹéĞŇæĹŚäžňä¡£çŤĺDALI reader åĞŕåřŚCPU æŢřæ■őåď ĎçŘĘèť§æŃĚåŠŇæŢřæ■őèŕżåŔŰçŞűéćĹã

```
paddle.enable_static()
configs = X.parse_train_configs()
fleet.init(is_collective=True)

model = X.applications.Resnet50()
downloader = X.utils.Downloader()
local_path = downloader.download_from_bos(
    fs_yaml='https://fleet.bj.bcebos.com/test/loader/small_imagenet.
↪yaml',
    local_path='./data')
batch_size = 32
loader = model.get_train_dataloader(local_path, batch_size=batch_
↪size)
```

10.3.4 åőŽäźĽåĹĘåÿČåijŕçŻÿåĚşç■ŰçŢě

è£ŹéĞŇæĹŚäžňäijŽåijĂåŘŕäÿŁæŰĞäÿ■æŔŘåĹřçŽĎåŘĎéąźèő■çżČäijŸåŇŰç■ŰçŢěïijŇåęĆïijŽèĞłåĿ
èđ■åŘĹç■L'ãĂĆ

```
dist_strategy = fleet.DistributedStrategy()

# distributed strategy
dist_strategy.sync_nccl_allreduce = True
dist_strategy.nccl_comm_num = 2
dist_strategy.fuse_all_reduce_ops = True
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
# build strategy
build_strategy = fluid.BuildStrategy()
build_strategy.enable_sequential_execution = True
build_strategy.fuse_elewise_add_act_ops = True
build_strategy.fuse_bn_act_ops = True
build_strategy.enable_auto_fusion = True
build_strategy.fuse_all_optimizer_ops = True
dist_strategy.build_strategy = build_strategy


# execute strategy
execution_strategy = fluid.ExecutionStrategy()
execution_strategy.num_threads = 3
execution_strategy.num_iteration_per_drop_scope = 100
execution_strategy.num_iteration_per_run = 1
dist_strategy.execution_strategy = execution_strategy

# amp
dist_strategy.amp = True
dist_strategy.amp_configs = {
    "init_loss_scaling": 128,
    "decr_every_n_nan_or_inf": 2,
    "incr_every_n_steps": 1000,
    "incr_ratio": 2.0,
    "use_dynamic_loss_scaling": True,
    "decr_ratio": 0.5,
    "custom_white_list": [],
    "custom_black_list": [],
}

dist_strategy.save_to_prototxt("dist_strategy.prototxt")
```

10.3.5 åijĂåğŃèő■çżČ

```
optimizer = fluid.optimizer.Momentum(learning_rate=0.01, momentum=0.
↪9)
optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)

place = fluid.CUDAPlace(int(os.environ.get('FLAGS_selected_gpus',␣
↪0)))
exe = fluid.Executor(place)
exe.run(fluid.default_startup_program())

for i, data in enumerate(loader()):
    start_time = time.time()
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
    cost_val = exe.run(model.main_prog,
                       feed=data,
                       fetch_list=[model.loss.name])

    end_time = time.time()
    print(
        "worker_index: %d, step%d cost = %f, speed: %f"
        % (fleet.worker_index(), i, cost_val[0], batch_size / (end_
↪time - start_time)))
```

10.3.6 Fleetrun äÿĂéŤőåŘŕåŁÍ

"xx.xx.xx.xxâĂİ ç■L'æŸŕåŻŻäÿłèŁĆçĆźipsïijŇæŕŔäÿłèŁĆçĆź 8 åijă GPUå■ąïijŇ åĚś 32 GPU åźűèąŇèő■çżČãĂĆ

```
fleetrun --ips="xx.xx.xx.xx, yy.yy.yy.yy, aa.aa.aa.aa, bb.bb.bb.bb"␣
↪--gpus=0,1,2,3,4,5,6,7 example_collective.py

# worker_index: 0, step0 cost = 7.147776, speed: 34.481360
# worker_index: 0, step1 cost = 7.151375, speed: 408.405991
# worker_index: 0, step2 cost = 7.025396, speed: 509.624355
# worker_index: 0, step3 cost = 6.501647, speed: 533.641315
# worker_index: 0, step4 cost = 6.759287, speed: 520.999193
# worker_index: 0, step5 cost = 6.266363, speed: 536.729215
# worker_index: 0, step6 cost = 6.243353, speed: 522.510241
# worker_index: 0, step7 cost = 6.923586, speed: 519.478763
# worker_index: 0, step8 cost = 7.607512, speed: 534.919526
# worker_index: 0, step9 cost = 7.111218, speed: 508.371600
```

10.4 Fleet èő■çżČç■ŰçŢě

10.4.1 DistributedStrategy

DistributedStrategy	çśżåđŃ	éżŸèőď	åĂ¿ æŞŔèę ĄèČ
auto	bool	False	èĞłåŁí åŇŰæąĘæđűåŔĆæŢřäijŸåŇŰ
a_sync	bool	True	æŇĞçď'ž æŸŕåŘęä¡£çŤĺåijĆæ■ěSGD è£Żèą̌ŇåŔĆ æŢřæŻť æŰřïijŇäżĚåIJÍPS serveræĺąåijŔäÿ■çŤ§æŢĹ
sync_nccl_allreduce	bool	True	æŇĞçď'žæŸŕ åŘęåIJĺæŕŔäÿłéĂŽä£ąçž£ çĺŃäÿ■äÿ■ä¡£çŤĺåŘŇæ■ě allre duceïijŇäżĚåIJÍColle ctiveæĺąåijŔäÿ■çŤ§æŢĹ ïijŇéĂŽåÿÿåIJĺä¡£çŤĺåŘŇ æ■ěallreduceåŘŐçşż çž§çŽĎåijĂéŤĂäijŽéŹ■ä¡Ő
nccl_comm_num	int	1	ncclé ĂŽä£ąåŹĺæŢřéĞŔ. ncclé ĂŽä£ąåŹĺæŢřéĞŔ nccl_comm_num åŔŕäżěåŁäå£ńGPUäźŃ éŮť çŽĎéĂŽä£ąæŢĹçŐĞïijŇ åżžèőőå■ŢæIJžèő¿ç¡őäÿž 1ïijŇåď'ŽæIJžèő¿ç¡őäÿž2 ãĂĆéŠĹåŕźCPUçž£çĺŃæŢř num_threads ïijŇåżžèőőå■ŢæIJžèő¿ç¡ő äÿž1ïijŇåď'ŽæIJžèő¿ç¡őäÿž nccl_comm_num +1
use_hierarchical_allreduce	bool	False	åĹĘçžğåijŔallred uceïijŇåŕźäžŐåď'ŽæIJžæĺą åijŔïijŇéŠĹåŕźåřŔæŢřæ■ő éĞŔçŽĎéĂŽä£ąïijŇRing AllReduc eéĂŽä£ąæŢĹçŐĞä¡ŐïijŇéĞĞ çŤĺHierarchical AllReduceåŔŕ äżěèğčåĘşèŕěéŮőéćŸãĂĆ
hierarchical_allreduce_inter_nranks	int	1	åIJÍ åĹĘçžğåijŔallreduc eïijŇä¡ŐåśĆçžğgroups äÿ■çŽĎ r ankæŢřãĂĆäÿĂèĹňç■Ľäžő å■ŢäÿłGPUèŁĆçĆźäÿ■çŽĎ GPUæŢř
sync_batch_norm	bool	False	èąĺçď'žæŸŕåŘęä¡£ çŤĺåŘŇæ■ěçŽĎæĽźæ■čåĹŹ åŇŰïijŇå■şåIJĺèő■çżČéŸű æőţéĂŽè£Ğåď'Žäÿłèő¿åď'Ğ ãĂĆå¡ŞåĽ■çŽĎåőđçŐřäÿ■ æŤŕæŇĄFP16èő■çżČåŠŇC PUãĂĆåźűäÿŤçŽőåĽ■* *äżĚæŤŕæŇĄ*äżĚåIJÍ äÿĂåŔřæIJžåŹĺäÿŁè£ŻèąŇ åŘŇæ■ěåijŔæĽźæ■čåĹŹãĂĆ
fuse_all_reduce_ops	bool	True	éżŸèőď æČĚåĘţäÿŃäijŽ åřĘåŘŇäÿĂlayeräÿ■åŔĆ æŢřçŽĎæćŕåžęçŽĎAllR educeæŞ■ä¡IJåŘĹåźűæĹŘ äÿĂäÿłïijŇæŕŤåęĆåŕźäžŐ fluid.layers.fc äÿ■æIJĽ WeightåŠŇBi asäýď äÿłåŔĆæŢřïijŇæĽŞåijĂèŕě éĂĽéąźäźŃåŘŐïijŇåŐ§æIJň éIJĂèęĄäýď æňąAllRed uceæŞ■ä¡IJïijŇçŐřåIJĺåŘł çŤĺäÿĂæňąAllReduce æŞ■ä¡IJãĂĆ
fuse_grad_size_in_MB	int	32	æŕŔäÿłAllReduceæŞ■ ä¡IJçŽĎæćŕåžęå■ŮèŁĆæŢř
fuse_grad_size_in_TFLOPS	int	20	æŇĞ åőŽæŕŔæňąAllReduc eæŞ■ä¡IJçŽĎæIJĂåď'ğåśĆæŢř ïijŇå■şåĹřè¿¿èŕěåśĆæŢř åřśè£ŻèąŇAllReduce
cudnn_exhaustive_search	bool	True	èąĺçď'žæŸŕ åŘęä¡£çŤĺçľ'ůäÿ¿æŘIJçť'ć æŰźæşŢæİěéĂĽæŃľåŕůçğŕ çőŮæşŢãĂĆåIJÍcuDNNäÿ■ æIJĽäýď çğ■æŘIJçť'ćæŰźæşŢ ïijŇåŘŕåŔŚåijŔæŘIJçť'ćåŠŇ çľ'ůäÿ¿æŘIJçť'ćãĂĆçľ'ůäÿ¿ æŘIJçť'ćåřİèŕŢæĽĂæIJĽcu DNNçőŮæşŢäżěéĂĽæŃľåĚű äÿ■æIJĂå£ńçŽĎçőŮæşŢãĂĆ æ■d'æŰźæşŢéİđåÿÿèĂŮæŮű

10.4.2 BuildStrategy

BuildStrategy	çśżåđ	ŕŹŸè	őžŽăźĽ
enable_sequential_execution	bool	False	åęĆæđIJ èő¿ç¡őäÿžTrueïijŇåĹŹ çőŮå■ŘçŽĎæĽğèąŇéąžåžŔ åřĘäÿŐçőŮå■ŘåőŽäźĽçŽĎ æĽğèąŇéąžåžŔçŻÿåŘŇãĂĆ
fuse_elewise_add_act_ops	bool	False	èąĺæŸŐ æŸŕåŘęèđ■åŘĹ(fuse) elementwise_add _opåŠŇactivation _opãĂĆè£ŹäijŽä¡£æŢť ä¡Ş æĽğèąŇè£ĞçĺŃæŻť å£ńãĂĆ
fuse_bn_act_ops	bool	False	èąĺæŸŐ æŸŕåŘęèđ■åŘĹ(fuse) batch_norm_op åŠŇ activation _opãĂĆè£ŹäijŽä¡£æŢť ä¡Ş æĽğèąŇè£ĞçĺŃæŻť å£ńãĂĆ
fuse_relu_depthwise_conv	bool	False	èąĺæŸŐ æŸŕåŘęèđ■åŘĹ(fuse) reluåŠŇ depthwise_conv 2dïijŇèŁĆçIJĄGPUåĘĚå■Ÿ åźűåŔŕèČ¡åŁăéĂ§æĽğèąŇ è£ĞçĺŃãĂĆæ■d'éĂĽéąźäżĚ éĂĆçŤĺäžŐGPUèő¿åd'ĞãĂĆ
fuse_broadcast_ops	bool	False	èąĺæŸŐ æŸŕåŘęèđ■åŘĹ(fuse) broadcast opsãĂĆ èŕěéĂĽéąźæŇĞåIJĺReduc eæĺąåijŔäÿŃæIJĽæŢĹïijŇä¡£ çĺŃåžŔè£ŘèąŇæŻť å£ńãĂĆ
fuse_all_optimizer_ops	bool	False	èąĺæŸŐ æŸŕåŘęèđ■åŘĹ(fuse) æŸŕåŘęèđ■åŘĹ optimizer_opïijŇäžĚåŕźéČĺåĹĘ optimizer åŔŕçŤĺ ïijĹSGDãĂĄAdamåŠŇM omentumïijĽïijŇåŔŕä¡£ çĺŃåžŔè£ŘèąŇæŻť å£ńãĂĆ
enable_inplace	bool	False	èąĺæŸŐæŸŕ åŘęOpçŽĎè¿ŞåĞžåd'■çŤĺO pè¿ŞåĚěçŽĎæŸ¿å■ŸçľžéŮť ïijŇäijŸåŇŰæŸ¿å■Ÿå■äçŤĺ
enable_backward_optimizer_op_deps	bool	True	åIJĺåŔ■åŘŚæŞ■ä¡IJ åŠŇåŔĆæŢřæŻť æŰřæŞ■ä¡IJ äźŃéŮť æůżåŁăä¿Ìèţű ïijŇ ä£Ìèŕ ĄåIJĺæĽĂæIJĽçŽĎåŔ■ åŘŚæŞ■ä¡IJéČ¡è£ŘèąŇçżŞ æİ§äźŃåŘŐæĽ■äijĂåğŃè£Ř èąŇåŔĆæŢřæŻť æŰřæŞ■ä¡IJ. åIJĺ åd'Žå■ąèőŋçżČæŮűïijŇæĽŞ åijĂèŕěéĂĽéąźåŔŕèČ¡äijŽ æŘŘå■Ğèő■çżČéĂ§åžęãĂĆ
cache_runtime_context	bool	False	unkown

10.4.3 ExecutionStrategy

Execution-Strategy	çśżåd	[illegible]	[illegible]
num_threads	int	1	èąĺçd'žå¡ŞåL'■ Executor çŽĎçž£çĺŃæśă(thread pool)çŽĎåd'ğářŕ, æ■d'çž£ çĺŃæśăåŕŕçŤĺæİěåźűåŕŠ æL'ğèąŇprogramäÿ■çŽĎ operatorïijĹçőŮå■Ř ïijŇè£ŘçőŮïijL'ãĂĆåęĆæđIJ num_threads=1 ïijŇåĹŹæL'ĂæIJL' çŽĎoperatorářĘäÿĂ äÿłæŐěäÿĂäÿłåIJřæL'ğèąŇ ïijŇä¡ĘåIJĺäÿ■åŘŇçŽĎpro graméĞ■åd'■åŚĺæIJ§(it erations)äÿ■æL'ğèąŇ éąžåžŕåŕŕèČ¡äÿ■åŘŇãĂĆ
num_iteration_per_drop_scope	int	10	èŕěéĂL'éąźèąĺ çd'žéŮt'éŽŤåd'ŽåřŚæňąè£■ äžčäźŃåŘŐæÿĚçŘĘäÿĂæňą äÿt'æŮűåŔŸéĞŘãĂĆæĺąåđŃ è£ŘèąŇè£ĞçĺŃäÿ■ïijŇçŤ§ æĹŘçŽĎäÿ■éŮt'äÿt'æŮűåŔŸ éĞŘåřĘèćńæŤ¿åĹřlocal execution scopeäÿ■ïijŇäÿžäžĘ éĄ£åĚ■åŕźäÿt'æŮűåŔŸéĞŘ éćŚçźĄçŽĎçŤşèŕůäÿŐéĞŁ æŤ¿ïijŇéĂŽåÿÿåřĘåĚűèő¿ äÿžè¿Čåd'ğçŽĎåĂijïijĹæŕŤ åęĆ10æĹŰèĂĚ100ïijL'ãĂĆ
num_iteration_per_run	int	3	åőČéĚ■ç¡őäžĘå¡ŞçŤĺæĹů åIJĺpythonèĎŽæIJñäÿ■ èřČçŤĺpe.run()æŮű æL'ğèąŇåŹĺäijŽæL'ğèąŇçŽĎ è£■äžčæňąæŢřãĂĆExecutoræŕŘæňąèřČçŤĺïijŇäijŽ è£ŻèąŇnum_iterat ion_per_runæňąèő■ çżČïijŇåőČäijŽä¡£æŢt'ä¡Ş æL'ğèąŇè£ĞçĺŃæŻt'å£ńãĂĆ
use_thread_barrier	bool	False	å¡Şä¡£çŤĺ PServer æĹąåijŘæŮűäÿž True

CHAPTER 11

èĞłåŁÍæůůåŘĹçš¿åžęçżČåŁăéĂ§åĹĘåÿČåijŕèő■çżČ

11.1 çőĂäżŃ

åIJÍä¡£çŤÍæŢřæ■őåźűèąŇåĹĘåÿČåijŕèő■çżČçŽĎåŘŇæŮű, æĹŚäžňè£ŸåŘŕäžěåijŢåĚèèĞłåŁÍæůůåŘĹçš¿åžę(Auto Mixed Precision) æİěè£ŻäÿĂæ■ěæŔŘå■Ğèő■çżČçŽĎéĂ§åžę.

äÿżæţĄçŽĎçěđçżŔç¡ŚçżIJæÍąåđŃéĂŽåÿÿä¡£çŤÍå■Ţçš¿åžę `single-precision (FP32)` æŢřæ■őæăijåijŔæİěå■ŸåĆÍæÍąåđŃåŘĆæŢřãĂĄè£ŻèąŇèő■çżČåŠŇéćĎæţŃ. åIJÍäÿŁè£řçŐŕèŁĆäÿ■ä¡£çŤÍå■Łçš¿åžę `half-precision (FP16)`æİěäžčæŻ£å■Ţçš¿åžę. åŘŕäžěåÿęæİěäžěäÿŃåě¡åď Ď:

1. åĞŔåřŚåŕźGPU memory çŽĎéIJĂæśĆ: GPU æŸ¿å■Ÿäÿ■åŔŸæČĚåĘţäÿŃ, æŤŕæŇĄæŻťåď ğæÍąåđŃ / batch size
2. éŹ■ä¡ŐæŸ¿å■Ÿèŕżå ĘŹæŮűçŽĎåÿęåő¡åŐŃåŁŻ
3. åŁăéĂ§GPU æŢřå■ęè£ŘçőŮéĂ§åžę (éIJĂèęĄGPU æŤŕæŇĄ[1])
4. GPUäÿŁ FP16 åŘđåŘŘæŸŕFP32 çŽĎ 2 - 8 åĂ■[2]

Paddle æŤŕæŇĄèĞłåŁÍæůůåŘĹçš¿åžęèőąçőŮ, åźűåőđçŐřäžĘ `èĞłåŁÍçżt'æŁd'FP32 ãĂĄFP16åŔĆæŢřåL'ŕæIJň`, `Dynamic loss scaling`, `opéżŚçŹ¡åŘ■å■Ţ` ç■Ľ'ç■ŰçŢěæİěéĄ£åĚ■ åŻă FP16 åŁÍæĂĄèŇČåŻť'è¿ČåřŔèĂŇåÿęæİěçŽĎæÍąåđŃæIJĂçżĹçš¿åžęæ■§åď śãĂĆ Fleet ä¡IJäÿžPaddleéĂŽçŤÍçŽĎåĹĘåÿČåijŕèő■çżČAPIæŔŘä¿ŻäžĘçőÅå■ŢæŸŞçŤÍçŽĎæŐěåŔč, çŤÍæĹůåŔłéIJĂèęĄæůżåŁăăĞăèąŇäžčçăĄ åŕśåŘŕåŕĘèĞłåŁÍæůůåŘĹçš¿åžęåžŤçŤÍåĹřåŐ§æIJL'çŽĎåĹĘåÿČå

äÿŃæŰĞårĘéĂŽè£ĞäÿĂäÿłçőÅå■Ţä¿Ńå■ŘäžŃçż■åęĆåęĆä¡ŢéĂŽè£Ğ FleetåřĘåőđçŐřæůůåŘĹçš¿åžęçŽĎåĹĘåÿČåijŕèő■çżČ, åŔęåď ŰçżŹåĞžæĹŚäžňä¡£çŤÍ Fleet è£ŻèąŇåŘŇæ■ěèő■çżČåŁăéĂ§çŽĎåőđèůţãĂĆ

11.1.1 èŕŢéłŇæŢĹædlJ

çŐŕåćČ: 4 æIJž 32å■ą V100-32GB

imagenet	å■Ţå■ą batch size	éĂ§åžę img/s	top1
‘VGG16-FP32	32	4133	55.4%
‘VGG16-AMP	32	7238	54.6%

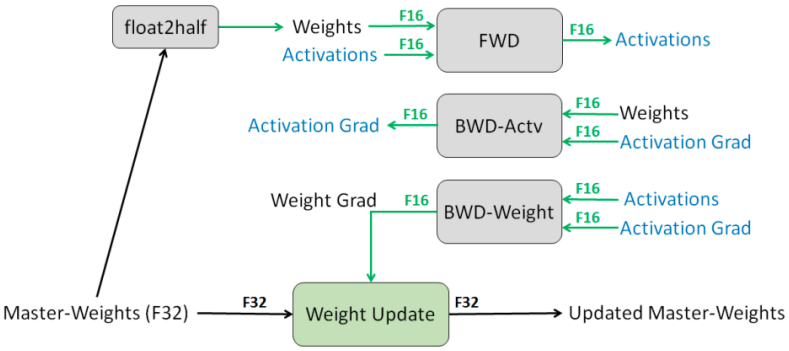
imagenet	å■Ţå■ą batch size	éĂ§åžę img/s	top1
‘Resnet50-FP32	128	8410	76.3%
‘Resnet50-AMP	128	25591	76.0%
‘Resnet50-FP32	256	OOM	OOM
‘Resnet50-AMP	256	29440	76.0%

11.2 AMP å£ńéĂ§åijÄåğŃ

è£ŹéĞŇäžěåIJĺå■ŢæIJžåď Žå■ąäÿŁèő■çżČResent50 äÿžçőÄå■Ţä¿Ńå■ŘäżŃçż■Fleet äÿ■AMPçŽĎçŤĺæşŢ.

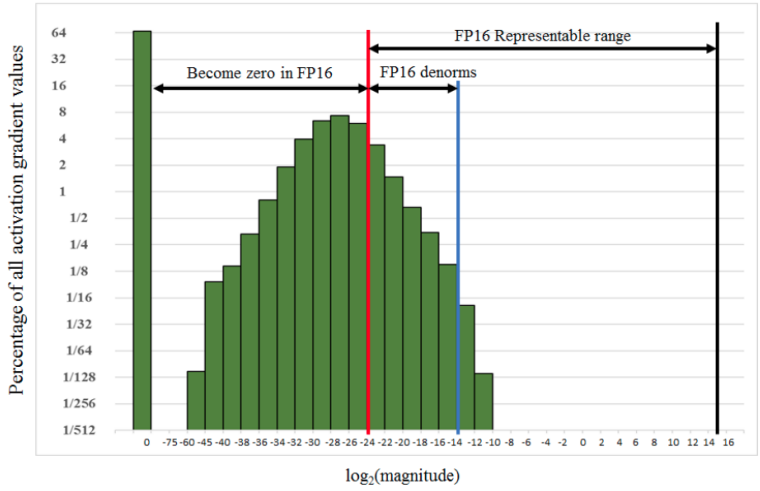
11.2.1 èĞłåŁĺæůůåŘĹçš¿åžęåŐ§çŘĘ

FP32 åŔĆæŢřåĽŕælJňåŔŁæŻťæŰř



åęĆäÿŁåŽ¿æL’Äçď ž, åIJĺAMP äÿ■, æĺąåđŃåŔĆæŢř `weight` , åL’■åŘŠäÿ■éŮť çŽĎçżŞæđIJ`activation`, åŔ■åŘŠçŽĎ`gradient` éČ¡äžěFP16 å¡ćåijŔå■ŸåĆĺ, çŤśæ■ď åŔŕäžěåĞŔåřŠæĺąåđŃå■ăçŤĺçŽĎæŸ¿å■ŸçľžéŮť ïijŇåŘŇæŮűæŔŘénŸèőąçőŮåŠŇé PaddleæąĘæđűäijŽäÿžæŕŔäÿĂäÿł`weight` çžť æŁď äÿĂäÿłFP32åĽŕæIJň, çŤĺäžŐåŔĆæŢřæŻť æŰř.

Loss scaling



åęĆäÿŁåŻ¿æL'Ăçd'ž, åőđéŹĚæČĚåĘţäÿ■æĺąåđŃèő■çżČäÿ■çŽĎæ§ŘäžŽåŔŸéĞŔ, æŕŤåęĆgrad (çL'źåĹńæŸŕ activation çŽĎ grad), åŔŕèČ¡äijŽåŻăářŔäžŐ FP16çŽĎçš¿åžęä¡ŐèĂŇåŔŸæĹŘ0;

åŔęäÿĂæŰźéIćåIJĺFP16 çŽĎèąĺçd'žèŇČåŻt'çŽĎäÿ■æIJL'å¿Ĺåd'ğçŽĎäÿĂéČĺåĹĘ(äžŐæIJĂåd'ğåĂijå¿Ă å■t'æšąæIJL'èćńåĹl'çŤĺåĹř.

åŕźgradient åĄŽäÿĂäÿłæŢt'ä¡ŞçŽĎæŤ¿åd'ğ, èČ¡åd'§æŻt'åĚĚåĹĘçŽĎåĹl'çŤĺFP16 çŽĎèąĺçd'žèŇČåŻt'.

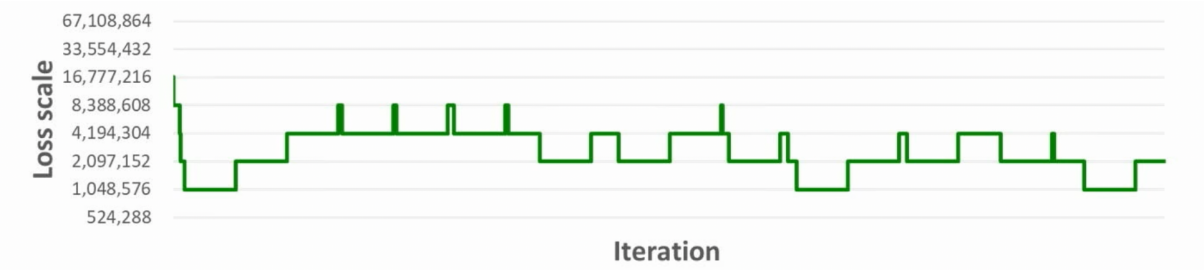
Fleet AMP äijŽåIJĺåŔ■åŘŠåijĂåğŃåL'■åŕź loss è£ŻèąŇ up scaling, åźűåIJĺæL'ğèąŇäżżä¡ŢæćŕåžęçŻÿåĚşæŞ■ä¡IJ(e.g. gradient-clip, update) äźŃåL'■åŕź gredient è£ŻèąŇ down scaling æĄćåd'■åŐ§æİěçŽĎåd'ğåřŔ.

scaling factor çŽĎèő¿ç¡őæŸŕ Lossing scaling çŽĎåĚşéŤő, Fleet AMP æŘŘä¿Ż Dynamic loss scaling ïijĹéżŸèőd'ïijL' åŠŇ Constant loss scaling äÿd'çğ■scaling ç■ŰçŢě:

- Constant loss scaling: èő¿ç¡ő use_dynamic_loss_scaling = False åŠŇ init_loss_scaling (float)
- Dynamic loss scaling: scaling äÿ■éĺcäÿt'çŽĎéŮőécŸæŸŕå¡Şscaling up äÿ■èűşæŮű, äż■äijŽæIJL'éČĺåĹĘè¿ČåřŔåŘŸéĞŔäijŽècńèąĺçd'žæĹŘ 0èĂŇæ■§åd'śçš¿åžę; å¡Şscaling up è£Ğåžę æŮű, åŘŸéĞŔèűĚè£ĞFP16èąĺçd'žèŇČåŻt'åĞžçŐř nan or inf, åŘŇæăůéĂăæĹŘçš¿åžęæ■§åd'ś. æ■d'ç■ŰçŢěéĞĞçŤĺèĞłåŁĺ gradient åĂijæčĂæţŃçŽĎæŰźåijŔ:
  - å¡Şè£đçż■incr_every_n_steps(int) äÿłbatch äÿ■æL'ĂæIJL'çŽĎ-gradient éČ¡åIJĺFP16 çŽĎèąĺçd'žèŇČåŻt', åřEscaling factor åćđåd'ğincr_ratio(float) åĂ■;
  - å¡ŞæIJL'è£đçż■decr_every_n_nan_or_inf(int) äÿłbatch äÿ■gradient éĞŇåĞžçŐř nan / infæŮű, scaling factor çijl'åřŔ decr_ratio(float) åĂ■.
  - äÿŁè£řåŻŽäÿłåŔĆæŢřFleet æŘŘä¿ŽçŽĎéżŸèőd'åĂijåŘŕäžěæżąęűşçżĺåd'ğéČĺåĹĘèęĄæśĆ,

çŤĺæĹůéĂŽåÿÿäÿ■éIJĂèęĄä£őæŤź.

åęĆäÿŃåŻ¿æĽĂçd'žåIJĺ Dynamic loss scaling äÿ■ïijŇæąĘæđűåIJĺæŕŔäÿĂäÿł iteration éČ¡äijŽä¿İæ■őå¡ŞåĽ■ gradients æŸŕåŘęåĞžçŐř `nan` or `inf` è£ŸæIJĽçŤĺæĹůèő¿ç¡őçŽĎ Dynamic loss scaling åŔĆæŢřæİěåŁĺæĂĄèřČæŢt' loss scaling factor çŽĎåd'ğåřŔijŇårĘgradient år¡éĞŔä£İæŇĄåIJĺ FP16 çŽĎèąĺçd'žèŇČåŻt'äźŃåĘĚãĂĆ



OP éżŚçŹ¡åŘ■å■Ţ

æĺąåđŃäÿ■çŽĎæ§ŘäžŻOperation (OP) åŕŕèČ¡åŕźçš¿åžęè¿ČäÿžæŢŕæĎ§, äÿžäžĘçąőä£İAMP äÿ■çš¿åžęæŮăæ■§, åŕŕäżěéĂŽè£ĞOP éżŚçŹ¡åŘ■å■Ţåŕźåĕůä¡ŞOP æŞ■ä¡IJçŽĎçš¿åžęåĄŽæŇĞåőŽ.

- çŹ¡åŘ■å■Ţ: OP æŞ■ä¡IJåIJĺFP16çš¿åžęäÿŃè£Żèą Ň, `input`: åęĆæđIJäÿ■æŸŕFP16 äijŽèćńéęŰåĚĹcast æĹŘFP16åŘŐåE■è¿ŞåĚěOP. `output`: FP16
- éżŚåŘ■å■Ţ: OP æŞ■ä¡IJåIJĺFP32çš¿åžęäÿŃè£Żèą Ň, `input`: åęĆæđIJäÿ■æŸŕFP32 äijŽèćńéęŰåĚĹcast æĹŘFP32åŘŐåE■è¿ŞåĚěOP. `output`: FP32
- çĄřåŘ■å■Ţ: æĽĂæIJĽäÿ■åIJĺéżŚæĹŰçŹ¡åŘ■å■ŢéĞŇçŽĎOP. äžĚå¡ŞOP æĽĂæIJĽ inputs éČ¡æŸŕ FP16çš¿åžęæŮű, æŞ■ä¡IJæĽ■åIJĺFP16çš¿åžęäÿŃè£Żèą Ň, åŘęçİĂäžěFP32è£Żèą Ň. `input / output`: åŠŇåŐ§åğŃè¿ŞåĚěäÿ■çŽĎæIJĂéńŸçš¿åžęçŻÿåŘŇ

Fleet åůšçżŘéćĎèő¿äžĘäÿĂäÿłèČ¡åd'§èęĘçŻŰçżİåd'ğåd'ŽæŢřæĺąåđŃOPsçŽĎéżŚçŹ¡åŘ■å■Ţ, éĂŽåÿÿæČĚåĘţäÿŃçŤĺæĹůåźűäÿ■éIJĂèęĄä£őæŤź, ä¡ĘæŸŕåęĆæđIJäżžåŁąåŕźçš¿åžęæIJĽçĽźæőŁèęĄæśĆ, æĹŰèĂĚåÿŇæIJŻæŰřåćđèĞłåőŽäźĽ OP, çŤĺæĹůåŘŕäžěéĂŽè£Ğ paddle.distributed.fleet.DistributedStrategy.amp_configs äÿ■çŽĎ `custom_white_list` åŠŇ `custom_black_list` è£ŻèąŇæŇĞåőŽ. åŘŇæŸŕ, çŤĺæĹůè£ŸåŘŕäžěéĂŽè£Ğ`custom_black_varnames`, æİěåĚůä¡ŞæŇĞåőŽ`Paddle program` æ§ŘäÿĂäÿł `var`å£Ěéąžä¡£çŤĺFP32çš¿åžę.

æĹŚäžňåřĘåIJĺæŰĞæIJńçŽĎ appendixäÿ■ è£ŻäÿĂæ■ěäžŃçż■ Fleet çŽĎéżŚçŹ¡åŘ■å■Ţèő¿ç¡őåŔŁåĚűå¡śåŞ■ãĂĆ

11.2.2 åijĂåğŃèő■çžČ

æůžåŁää¿İètŰ

éęŰåĚĹæĹŚäžňèęĄåŕijåĚëä¿İètŰåŠŇåőŽäźĽæĺąåđŃåŠŇ data loader, è£ŹäÿĂæ■ěåŠŇFleet äÿŃåĚűäžŰäżžåŁąå§žæIJňäÿĂèĞť.

```
import os
import fleetx as X
import paddle
import paddle.fluid as fluid
import paddle.distributed.fleet.base.role_maker as role_maker
import time
import paddle.distributed.fleet as fleet
```

åőŽäźĽåĹĘåÿČåijŕæĺąåijŕåźűåĹlåğŃåŇŰ

```
paddle.enable_static()
configs = X.parse_train_configs()
fleet.init(is_collective=True)
```

åŁăè¡¡æĺąåđŃåŕŁæŢřæ■ő

```
model = X.applications.Resnet50()
downloader = X.utils.Downloader()
local_path = downloader.download_from_bos(
    fs_yaml='https://fleet.bj.bcebos.com/test/loader/small_imagenet.
↪yaml',
    local_path='./data')
batch_size = 32
loader = model.get_train_dataloader(local_path, batch_size=batch_
↪size)
```

åőŽäźĽåĹĘåÿČåijŕåŕŁAMP çŻÿåĚşç■ŰçŢě

åęĆäÿŁæŰĞæŕŕè£ř, çŤĺæĹůåŕŕäžěéĂĽ'æŃl'èő¿ç¡ő `Loss scaling` åŠŇ `OPéżŚçŹ¡åŘ■å■Ţç■L'çŽĎåŕĆæŢř.

åŕęåd'Ű Fleet åřĘAMP åőđçŐřäÿž meta optimizer, çŤĺæĹůéIJĂèęĄæŇĞåőŽåĚűçŽĎ `inner-optimizer`. Fleet AMPæŤŕæŇĄæĽ'ĂæIJL' paddle optimziers åŠŇ FLeet meta optpimizers ä¡IJäÿžåĚű inner-optimizer.

```
dist_strategy.amp = True
dist_strategy.amp_configs = {
    "init_loss_scaling": 32768,
    "decr_every_n_nan_or_inf": 2,
    "incr_every_n_steps": 1000,
    "incr_ratio": 2.0,
    "use_dynamic_loss_scaling": True,
    "decr_ratio": 0.5,
    "custom_white_list": [],
    "custom_black_list": [],
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
}

optimizer = fluid.optimizer.Momentum(learning_rate=0.01, momentum=0.
↪9)
optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)
```

åijĂåğŃèő■çżČ

è£ŹäÿĂéČÍåĹĘåŠŇFleet äÿ■åĔűäžŰäżżåŁąå§žæIJňçŻÿåŘŇ:

```
place = fluid.CUDAPlace(int(os.environ.get('FLAGS_selected_gpus',␣
↪0)))
exe = fluid.Executor(place)
exe.run(fluid.default_startup_program())

for i, data in enumerate(loader()):
    start_time = time.time()
    cost_val = exe.run(model.main_prog,
                       feed=data,
                       fetch_list=[model.loss.name])

    end_time = time.time()
    print(
        "worker_index: %d, step%d cost = %f, speed: %f"
        % (fleet.worker_index(), i, cost_val[0], batch_size / (end_
↪time - start_time)))
```

11.2.3 è£ŘèąŇèő■çżČèĎŽæIJň

äÿĂèąŇåŘŕåŁĺå■ŢæIJžåďŽå■ąåĹĘåÿČåijŔèő■çżČïijŽ

```
fleetrun --gpus 0,1,2,3,4,5,6,7 --log_dir log example_amp.py

# worker_index: 0, step0 cost = 6.895311, speed: 12.192901
# worker_index: 0, step1 cost = 6.964077, speed: 412.116618
# worker_index: 0, step2 cost = 7.049311, speed: 433.850506
# worker_index: 0, step3 cost = 7.006689, speed: 358.400410
# worker_index: 0, step4 cost = 7.000206, speed: 398.210745
# worker_index: 0, step5 cost = 7.088611, speed: 462.322357
# worker_index: 0, step6 cost = 7.022367, speed: 425.185013
```

11.2.4 Fleet éżŚçŹ¡åŘ■å■Ţèő¿ç¡ő

äÿŁæŰĞçőĂèęĄäżŃçż■äžĘFleet äÿ■éżŚçŹ¡åŘ■å■ŢçŽĎ API æŐěåŕčïijŇ äÿŃæŰĞåřĘè£ŻäÿĂæ■ěäżŃçż■ Fleet äÿ■éżŚçŹ¡åŘ■å■ŢçŽĎåőđçŐřåŠŇåŕřèČ¡åŕźèő■çżČéĂăæĹŘå¡śåŞ■ãĂ
çŻőåL'■ Fleet äÿ■ AMP çŽĎéżŸèőd'éżŚçŹ¡åŘ■å■ŢåęĆäÿŃïijŇ åĚűäžŰæIJłåĹŮåĞžçŽĎ op éČ¡åśđäžŐçĄřåŘ■å■ŢïijŽ

```
white_list = {
    'conv2d',
    'matmul',
    'mul',
}
black_list = {
    'exp',
    'square',
    'log',
    'mean',
    'sum',
    'cos_sim',
    'softmax',
    'softmax_with_cross_entropy',
    'sigmoid_cross_entropy_with_logits',
    'cross_entropy',
    'cross_entropy2',
}
```

éżŚçŹ¡åŘ■å■Ţèő¿ç¡ő

çŹ¡åŘ■å■Ţäÿ■åŔłæIJL'å■ůçğŕåŠŇäźŸæşŢè£ŘçőŮïijŇè£ŹæăůçŽĎèő¿ç¡őèČ¡åd'§æżąèűşåd'ğéČĺåĹĘçŽĎ CV åIJžæŹŕçŽĎæĺąåđŃåŁăéĂ§ïijĹVggãĂĄResNetïijL'ïijŇ åŻääÿžå■ůçğŕèőąçőŮå■ăæ■őè£ŹäžŻæĺąåđŃèőąç åĚűäžŰ ops çŽĎåijĂéŤĂåŔłå■ăå¿ĹåřŔäÿĂéČĺåĹĘãĂĆ åŕźäžŐ äÿżèęĄåijĂéŤĂåIJÍ RNN èőąçőŮçŽĎ NLP æĺąåđŃïijŇçŻőåL'■çŽĎ AMP åőđçŐřæŔŘéĂ§åźűäÿ■æŸŕå¿ĹæŸŐæŸ¿ãĂĆ

éżŚåŘ■å■Ţäÿ■çŽĎ op åŔŕäžěåĹĘäÿž3 åd'ğçśżïijŽ * åŕźçš¿åžęéİđåÿÿæŢŔæĎ§çŽĎ opïijŽ `softmax`ïijŇ`cross_entropy` ç■L'ãĂĆ * è¿ŞåĞžçŻÿåŕźäžŐè¿ŞåĚěæIJL'æŻt'åd'ğåŁĺæĂĄèŇČåŻt'çŽĎopïijĹf(x) >> xïijL'ïijŽ`exp`ïijŇ`square`, `log` ç■L'ãĂĆ * reduce çśžåđŃçŽĎopïijŽ `mean`ïijŇ`sum` ç■L'ãĂĆ æL'ĂäżěïijŇç̀ĺæĹůåÿŇæIJŻåĹd'æŰ■æŰŕçŽĎèĞłåőŽäźL'opæŸŕåŘęéIJĂèęĄåŁăåĚěéżŚåŘ■å■ŢæŮűïijŇåŔŕ

éIJĂèęĄæşĺæĎŔ: äÿĂäžŻåÿÿçŤĺçŽĎ op åęĆ `BatchNorm`ïijŇ `pooling`ïijŇ `relu` åśđäžŐçĄřåŘ■å■ŢïijŇè£ŹæĎŔåŚşçİĂè£ŹäžŻ op çŽĎæŢřæ■őçśžåđŃåĘşåőŽäžŐäźŃåL'■çŽĎ op çŽĎçśżåđŃïijŻ åŘęåd'Ůåźűèą̀ŇåĹĘåÿČåijŔèőąçőŮä¡£çŤÍ AMPäźŃåŘŐïijŇgradient-allreduce æŸŕåIJÍFP16 äÿ■è£ŻèąŇçŽĎãĂĆ

èĞłåŁÍåŇŰop æŕŠåĚě

åIJÍèő■çżČåijĂåğŇåL'■ïijŇæąĘæđűäijŽæäźæ■őéżŠçŹ¡åŘ■å■ŢåIJÍåL'■åŘŠåŠŇåŕ■åŘŠç¡ŠçżIJèĞłåŁÍæŕcast opïijŇ åęĆïijŽ * åL'■åŘŠäÿ■æŕŠåĚě FP32toFP16 castïijŇ åřĘ FP32 çŽĎlayer parameter åL'ŕæIJň cast æĹŘ FP16ïijŇ è£Żèą Ň FP16 conv èőąçőŮãĂĆ * åŕ■åŘŠäÿ■æŕŠåĚě FP16toFP32 castïijŇ åřEç■L'åĹřçŽĎ FP16 gradient cast æĹŘ FP32ïijŇ çĎűåŘŐæŻt'æŰř FP32 çŽĎparameter åL'ŕæIJňãĂĆ

cast op èŹ¡çĎűäijŽåÿęæÌěéćÌåd'ŰçŽĎåijĂéŤĂïijŇ ä¡ĘæŸŕåIJÍèŕÿåęĆ VggãĂĄResNet ç■L'äÿżèęĄçŤšéĞ■åd'■çŽĎ conv layer äÿšèąŇçŽĎèĂŇæĹŘ CV æÍąåđŃäÿ■ïijŇ åŘłéIJĂèęĄcast input åŠŇ æŕŔäÿĂåśĆçŽĎparamïijŇåźűäÿ■éIJĂèęĄcast æÍąåđŃçŽĎäÿ■éŮt'çżŞæđIJïijŇè£Źæăů cast æŞ■ä¡IJåÿęæÌěçŽĎåijĂéŤĂè¿ČårŠ, åőźæŸŞåĂ■å■Łçš¿åžęèőąçőŮåÿęæÌěçŽĎåŁăéĂ§èęĘçŻŰïijŻä¡ĘæŸŕåęĆæđIJæÍąåđŃçŽĎäÿšèąŇ layers åžŔåĹŮäÿ■å■ŸåIJÍè¿Čåd'ŽçŽĎéżŠåŘ■å■Ţ opïijŁe.g. conv –> log –> conv –> square –> convïijL'ïijŇ è£ŹæăůæÍąåđŃçŽĎäÿ■éŮt'çżŞæđIJéIJĂèęĄè£ŻèąŇåd'Žæňą FP32toFP16 åŠŇ FP16toFP32 castïijŇ cast åijĂéŤĂårĘäijŽæĂěåL'ğåćđåd'ğïijŇäžŐèĂŇæŁţæűĹå■Łçš¿åžęåÿęæÌěçŽĎåŁăéĂ§ă

åŕŕèČ¡äÿ■éĂĆçŤÍ AMP åŁăéĂ§çŽĎæČĚåĘţ

- RNN äÿžäÿżçŽĎ NLP æÍąåđŃ
- æÍąåđŃçżĎç¡Śäÿ■æIJL'è¿Čåd'ŽéżŠåŘ■å■Ţ op çŽĎæÍąåđŃ
- åŕźæŢřæ■őçš¿åžęæŢŔæĎ§çŽĎäżžåŁąïijĹAdversarial Attacking in MLïijL'

åŻ¿åČŔ Input Layout æăijåijŕ

CV æÍąåđŃèő■çżČæŮűäžĘè¿¿åĹřæIJĂä¡şéĂ§åžęïijŇäÿ■åŘŇåIJžæŹŕäÿŇæŐÍè■Řä¡£çŤÍäÿ■åŘŇåŻ¿åČŔLayoutïijŽ

- FP32ïijŽNCHW
- èĞłåŁÍæůůåŘĹçš¿åžęïijŽ NHWC

```
# when build dataloader
loader = model.load_imagenet_from_file("./ImageNet/train.txt",
                                       batch_size=args.batch_size,
                                       data_layout="NHWC")

# when build model
if data_format == "NHWC":
    img_shape = [None, 224, 224, 3]
else:
    img_shape = [None, 3, 224, 224]
image = fluid.data( name="feed_image", shape=img_shape, dtype=
↪"float32", lod_level=0)
conv = fluid.layers.conv2d(input=input, data_format= "NHWC")
```

11.3 æŐÍè■ŘéŸĚèŕż:

åęĆæđIJéIJĂèęĄåŕźèĞłåŁÍæůůåŘĹçš¿åžęåĄŽåőŽåĹűåŇŰä£őæŤź,æĹŰæŻť æůśåĚěçŘĘèğčAMPäÿ■åŐ

- Mixed Precision Training
- MIXED PRECISION TRAINING: THEORY AND PRACTICE
- Training With Mixed Precision

CHAPTER 12

ä¡£çŤĺèűĚåď̆gBatchè£Žèą Ňèő■çżČ

12.1 çőĂäžŃ + strategyåĹÙèąĺ

äÿžäžĘè£¡æśĆæĺąåđŃçŽĎæĂğèČ¡äÿ■æŰ■æŔŘå■ĞïijŇäžžäžňářźæŻť̆åď̆ğèğĎæĺąçŽĎæŢřæ■őéŻĘãĂĄ

- è¿ŞåĚěçŽĎæŢřæ■őè£Ğåď̆ğïijŇä¿ŃåęĆèğĘéćŚçśżèő■çżČæŢřæ■őãĂĆ
- æůśåžęæĺąåđŃçŽĎåŔĆæŢřè£Ğåď̆ŽæĹŰè£Ğåď̆ğïijŇæL’ĂéIJĂçŽĎå■ŸåĆĺçľžéŮť̆èűĚåĞžäžĘåĘĚå■Ÿ
- AIèŁŕçL’ĞçŽĎåĘĚå■ŸæIJL’éŹŘãĂĆ

äÿžäžĘèČ¡æ■čåÿÿåőŇæĹŘèő■çżČïijŇæĹŚäžňéĂŽåÿÿåŔłèČ¡ä¡£çŤĺè¿ČåřŔçŽĎBatch SizeäžěéŹ■ä¡ŐæĺąåđŃèő■çżČäÿ■çŽĎæL’ĂéIJĂèęAçŽĎå■ŸåĆĺçľžéŮť̆ïijŇè£ŹåřĘåŕijèĞť̆å¿Ĺåď̆ŽæĺąåđŃæŰ SizeæİěæŔŘéńŸæĺąåđŃçŽĎçš¿åžęãĂĆäÿžäžĘèğčåĘşè£ŹäÿłéŮőécŸïijŇFleetäÿ■æŔŘä¿ŻäžĘäÿď̆çğ■ç■ŰçŢ

- **Forward Recomputation Backpropagation ïijĹFRBïijL’ïijŽ** éĂŽè£ĞæÿĚéŹď̆æ■čåŘŚèőąçőŮè£ĞçĺŃäÿ■çŽĎäÿ■éŮť̆èőąçőŮçżŞæđIJïijŇæİěéŹ■ä¡ŐèŐ■çżČè£ĞçĺŃ SizeçŽĎèő■çżČãĂĆ
- **Gradient MergeïijŽ** åIJĺèő■çżČè£ĞçĺŃäÿ■ïijŇåřĘè£đçż■åď̆ŽäÿłBatchæŢřæ■őèŐ■çżČå¿ŮåĹřçŽĎæćŕ

12.2 åŐ§çŘĘ

12.2.1 Forward Recomputation Backpropagation

æĹŚäžňç§ěéĄŞïijŇæůśåžęå■ęäźžăç¡ŚçżIJçŽĎäÿĂæňąèő■çżČè£■äžčåŇĚåŘńäÿL’äÿłæ■ěéłď̆ïijŽ

- **åL’■åŘŚèőąçőŮïijŽ** è£ŘèąŇåL’■åŘŚçőŮå■Ř(Operator) æİěèőąçőŮäÿ■éŮť̆éŽŘåśĆ(Variable)çŽĎåĂij ãĂĆ

- **åŔ■åŘŠèőąçőŮïijŽ** è£ŘèąŇåŔ■åŘŠçőŮå■ŘæİěèőąçőŮåŔĆæŢř(Parameter)çŽĎæćŕåžęãĂĆ
- **äijŸåŇŰïijŽ** åžŤçŤĺäijŸåŇŰçőŮæşŢäžěæŻťæŰřåŔĆæŢřåĂij ãĂĆ

åIJĺåL'■åŘŠèőąçőŮè£ĞçÍŇäÿ■ïijŇåL'■åŘŠçőŮå■ŘäijŽèőąçőŮåĞžåď'ğéĞŔçŽĎäÿ■éŮť'çżŞæđIJïijŇçŤśă
SizeèűŁåď'ğïijŇäÿ■éŮť'çżŞæđIJå■ăçŤĺçŽĎåĘĚå■ŸäźŞåřśèűŁåď'ğãĂĆéčđæąĺæäÿå£ČæąĘæđűäijŽä¡£çŤĺ
Variableæİěå■ŸåĆĺè£ŹäžŽéŽŘåśĆçŽĎäÿ■éŮť'çżŞæđIJãĂĆå¡ŞæĺąåđŇåśĆæŢřåŁäæůśæŮűïijŇåĚűäÿ■éŮť'çż
å■ăæ■őåď'ğéĞŔçŽĎåĘĚå■ŸãĂĆèŹ¡çĎűéčđæąĺæäÿå£ČæąĘæđűçŽĎæŸ¿å■ŸåŽđæŤűæIJžåĹűäijŽåŔŁæŮűæ
ä¡ĘæŸŕæIJL'äžŻäÿ■éŮť'çżŞæđIJæŸŕåŔ■åŘŠèőąçőŮè£ĞçÍŇäÿ■çőŮå■ŘçŽĎè¿ŞåĚěïijŇè£ŹäžŻäÿ■éŮť'çżŞæ

åŕźäžŐåď'ğåřŔåŽžåőŽçŽĎåĘĚå■Ÿæİěèŕť'ïijŇåęĆæđIJçŤĺæĹůåÿŇæIJŻä¡£çŤĺåď'ğBatch
SizeçŽĎæŢřæ■őè£ŻèąŇèő■çżČïijŇåĹŹåřĘåŕijèĞt'å■Ţäÿłäÿ■éŮť'çżŞæđIJå■ăçŤĺåĘĚå■Ÿåćđåď'ğïijŇéĆčäźĹå

FRBæŸŕåřĘæůśåžęå■ęäźăç¡ŚçżIJåĹĞåĹĘäÿžkäÿłéČĺåĹĘïijĹsegmentsïijL'ãĂĆåŕźæŕŔäÿłsegmentèĂŇèĹ

æĹŚäžňæŁŁåĹĞåĹĘç¡ŚçżIJçŽĎåŔŸéĞŔåŔńåĄŽcheckpointsãĂĆ
éĆčäźĹéŮőécŸæİěäžĘïijŇåęĆä¡ŢéĂL'æŃl'checkpointsåŚćïij§èĞłäžŐFRBæŰźæşŢæŔŘåĞžäžěæİěïijŇåď'ğé
æĹŚäžňç§ééĄŞæůśåžęå■ęäźăç¡ŚçżIJéĂŽåÿÿæŸŕçŤśäÿĂäÿłäÿłæĺąåİŮäÿšèĄŤå¿ŮåĹřçŽĎïijŇæŕŤåęĆResNet
50çŤś16äÿłblockäÿšèĄŤèĂŇæĹŘïijŇ Bert-LargeçŤś24äÿłtransformeräÿšèĄŤèĂŇæĹŘïijŇäžěäÿd'äÿłå■ŘæĺąĄ
åŕźäžŐéİđäÿšèĄŤçŽĎç¡ŚçżIJïijĹæŕŤåęĆåŔńæIJL'åď'ğéĞŔshortcutçżŞæđĎçŽĎç¡ŚçżIJïijL'ïijŇFRBäź§æŤræ
åŘłæŸŕåŔŕèČ¡åď'ŽèĂŮèť'źäÿĂçĆźåĘĚå■ŸïijĹçŤĺäžŐå■ŸåĆĺshortcutçŽĎVariableïijL'ãĂĆ

12.2.2 Gradient Merge

äÿŐFRBçŻÿæŕŤïijŇGradient Mergeåźűæšąæ IJL'åČŔFR-
BéĆčæăůåŕźåĘĚå■ŸçŽĎä¡£çŤĺåĄŽåĞžåď'ğåĹĂéŸŤæŰğèĹňçŽĎæŤźåŁĺïijŇåŘłæŸŕåIJÍèő■çżČæţĄçÍŃäÿŁå
Sizeèő■çżČæŢĹæđIJçŽĎçŽőçŽĎãĂĆåĚůä¡Şæİěèŕť'ïijŇåřśæŸŕä¡£çŤĺèŇěåźšåŐ§æIJL'åď'ğåřŔçŽĎBatchæŢŕä
ç¡ŚçżIJèőąçőŮå¿ŮåĹřæćŕåžęãĂĆåĚűéŮť'äijŽæIJL'äÿĂéČĺåĹĘæŸ¿å■Ÿ/åĘĚå■ŸçŤĺäžŐå■ŸæŤ¿æćŕåžęïijŇçĹ

GradientMerge ç■ŰçŢěåIJÍä¡£çŤĺæŰźéİćäź§å¿ĹçőĂå■ŢïijŇçŤĺæĹůåŘłéIJĂèęĄåőŽäźL'åřĘåď'ŽåřŚBatc

12.3 æŞ■ä¡lJåőđèůţ

èŕěçńäèŁĆäÿ■æĹŚäžňåřĘå§žäžŐBERTæĺąåđŇçŽĎåőđçŤĺæăůä¿ŃïijŇåĹĘåĹńåŕźè£Źäÿd'äÿłåćđåď'ğBat

- æůżåŁäèő■çżČèĎŽæIJňè£ŘèąŇæL'Ăå£ĚéąžçŽĎä¿İèţŰåŇĚãĂĆ
- åőŽäźL'åĹĘåÿČåijŔæĺąåijŔåźűåĹİåğŃåŇŰãĂĆ
- åŁăè¡¡æĺąåđŃåŔŁæŢřæ■őãĂĆ
- åőŽäźL'èő■çżČç■ŰçŢěåŠŇäijŸåŇŰåŹĺïijŇåIJÍè£ŹäÿĂæ■ěæĹŚäžňåŔŕäžěéĂL'æŃl'ä¡£çŤĺFRBæĹŰèĂ
  Mergeç■ŰçŢěæİěåćđåď'ğBatchSizeãĂĆ

äÿŃéİćæĹŚäžňæİěåĹĘåĹńäžŃçż■FRBåŠŇGradient Mergeäÿd'çğ■ç■ŰçŢěæL'ĂåŕźåžŤèĎŽæIJňçŽĎçijŰ
åŔŁ bert_gradient_merge.py)ãĂĆ

12.3.1 Forward Recomputation Backpropagation

æůżåŁää¿ièţŰ

éęŰåĚĹæĹŚäžňéIJĂèęĄæůżåŁäèő■çżČäÿ■æĽĂçŤĺåĹřçŽĎpythonæĺąåİŮïijŇfleetx åŕŕäžěçŤĺäžŐåŁäè¡¡æĹŚäžňäÿžçŤĺæĹůåřĄèčĚçŽĎæŐěåŔčåęĆïijŽåŁäè¡¡æĺąåđŃåŔŁæŢřæ■őïijŇæĺąåđŃèő■ distributed.fleet äÿ■åőŽäźĽäžĘäÿřářŇçŽĎåĹĘåÿČåijŔç■ŰçŢëä¿ŻçŤĺæĹůä¡£çŤĺãĂĆ

```
# -*- coding: UTF-8 -*-
import paddle
import fleetx as X
import paddle.fluid as fluid
import paddle.distributed.fleet as fleet
```

åőŽäźĽåĹĘåÿČåijŔæĺąåijŔåźůåĹiåğŃåŇŰ

éĂŽè£ĞX.parse_train_configs()æŐěåŔčijŇçŤĺæĹůåŕŕäžěåőŽäźĽèő■çżČçŽÿåĚşçŽĎåŔĆæŢ init()æŐěåŔčåőŽäźĽäžĘåĹĘåÿČåijŔæĺąåđŃïijŇäÿŃéİćäžčçăĄäÿ■çŽĎis_collective=Trueèąĺçd'ž

```
paddle.enable_static()
configs = X.parse_train_configs()
fleet.init(is_collective=True)
```

åŁäè¡¡æĺąåđŃåŔŁæŢřæ■ő

çŤĺæĹůåŕŕäžěéĂŽè£ĞX.applicationsæŐěåŔčåŁäè¡¡æĹŚäžňécĎåĚĹåőŽäźĽåě¡çŽĎæĺąåđŃïijŇåę SizeïijĹ130ïijĽæİèè£ŻèąŇèő■çżČãĂĆ

äÿŐæ■d'åŘŇæŮűïijŇçŤĺæĹůåŕŕäžěä¡£çŤĺæĹŚäžňçŽĎ'Downloader'æŐěåŔčäÿŃè¡¡écĎåĚĹä£İå■ŸçŽĎ

```
model = X.applications.BertLarge()
downloader = X.utils.Downloader()
local_path = downloader.download_from_bos(
    fs_yaml='https://fleet.bj.bcebos.com/small_datasets/yaml_
↪example/wiki_cn.yaml',
    local_path='./data')
data_loader = model.get_train_dataloader(
    local_path,
    max_seq_len=512,
    batch_size=130,
)
```

åőŽäźĽRecompute Strategy åŔŁ Optimizer

æŐěäÿŃæİěæĹŚäžňářšåŕŕäžěåőŽäźĽåĹĘåÿČåijŔèő■çżČäÿ■æĽĂåžŢçŤĺåĹřçŽĎç■ŰçŢëäžĘãĂĆäÿŃéİ recomputeèő¿ç¡őäÿžTrue åźűèő¿ç¡őæĹŚäžňäžŃåĚĹåőŽäźĽåě¡çŽĎcheckpointsãĂĆ

æŐěäÿŃæİęçŤĺæĹůéIJĂèęĄåőŽäźĽèő■çżČäÿ■æŽťæŰřæĺąåđŃæĽĂçŤĺåĹřçŽĎäijŸåŇŰåŹĺïijŇåźűä¡£ç distributed_optimizeræŐěåŔčářĘäijŸåŇŰåŹĺè¡ñæ■ćäÿžåĹĘåÿČåijŔæĺąåijŔãĂĆ

æIJĂåŘŐè£ŘèąŇoptimizer.minimize(model.loss)
åřĘåŔ■åŘŠèőąçőŮçŽĎçőŮå■ŘæŔŠåĚèè■çżČç¡ŚçżIJïijŇæĹŚäžňåřśåŔŕäžěåijĂåğŃèő■çżČäžĘãĂĆ

```
dist_strategy = fleet.DistributedStrategy()
# ä¡£çŤĺRecomputeïijŇåźűèő¿ç¡őcheckpoints
dist_strategy.recompute = True
dist_strategy.recompute_configs = {"checkpoints": model.checkpoints}

optimizer = fluid.optimizer.Adam(learning_rate=configs.lr)
optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)
```

åijĂåğŃèő■çżČ

åIJÍ FleetX äÿ■ïijŇæĹŚäžňäÿžçŤĺæĹůæŔŘä¿ŻäžĘX.MultiGPUTrainer
æŐěåŘčïijŇçŤĺäžŐGPUåĹĘåÿČåijŔèő■çżČãĂĆåĚűäÿ■model åŔŁ data_loader
åĹĘåĹńäÿžçňňäžŇæ■ěäÿ■åŁäè¡¡çŽĎæĺąåđŃåŔŁæŢřæ■őãĂĆstart_step
èąĺçd'žåijĂåğŃæL'Şå■řèő■çżČlogçŽĎæ■ěæŢřïijŇèŃěçŤĺæĹůæČşåd'■çŐřæĹŚäžňçŽĎæĺąåđŃèő■çżČéĂ§åžę

```
trainer = X.MultiGPUTrainer()
trainer.fit(model, data_loader, epoch=10)
```

è£ŘèąŇèő■çżČèĎŽæIJň

åőŇæĹŘèĎŽæIJňçŽĎçijŰåĘźåŘŐæĹŚäžňåřśåŔŕäžěä¡£çŤĺäžěäÿŃåŚ¡äžd'èő■çżČåĹĘåÿČåijŔæĺąåđŃïij

```
fleetrun --gpus 0,1,2,3,4,5,6,7 bert_recompute.py
```

æŢĹædIJæţŃèŕŢ

æĹŚäžňåIJÍBERTæĺąåđŃäÿŁåŕźrecomputeçŽĎæŢĹædIJè£ŻèąŇäžĘæţŃèŕŢïijŇä¡£çŤĺRecomputeåŘŐBa
sizeåŔŕäžěæL'l'åd'ğ9åĂ■åd'ŽãĂĆäÿŐæůůåŘĹçš¿åžęäÿĂèţůä¡£çŤĺæŮűïijŇBatch_sizeåŔŕäžěè£ŻäÿĂæ■ěæL'

Model	Baseline	Recompute	Recompute + mixed precision
Batch size	14	130	145
speed	69.92 sents/s	45.76 sents/s	75.84 sents/s

12.3.2 Gradient Merge

äÿŃéİćïijŇæĹŚäžňäžŃçż■åęĆä¡Ţä¡£çŤĺ Gradient Merge
æİěæL'l'åd'ğBERTæĺąåđŃåĹĘåÿČåijŔèő■çżČäÿ■çŽĎ Batch
SizeïijĹåĄĞèő¿èĎŽæIJňåŘ■çğřäÿžbert_gradient_merge.pyïijL'ïijŽ

äÿŐ Forward Recompute Backpropagation çŽÿåŘŃïijŇæĹŚäžňéęŰåĚĹèęĄæůżåŁää¿İèţŰïijŇåőŽäźL'åĹ

æůżåŁää¿İèţŰ

```
# -*- coding: UTF-8 -*-
import paddle
import fleetx as X
import paddle.fluid
import paddle.distributed.fleet as fleet
```

åőŽäźĽåĹĘåÿČåijŕæĺąåijŕåźűåĹlåğŃåŇŰ

åŁäè¡¡æĺąåđŃåŕŁæŢřæ■ő

```
model = X.applications.Bert_large()
downloader = X.utils.Downloader()
local_path = downloader.download_from_bos(
    fs_yaml='https://fleet.bj.bcebos.com/small_datasets/yaml_
↪example/wiki_cn.yaml',
    local_path='./data')
data_loader = model.(
    local_path,
    max_seq_len=512,
    batch_size=13,
)
```

åőŽäźĽGradient Merge Strategy åŕŁ Optimizer

åIJĺäÿŁéİćçŽĎäžčçăĄäÿ■ïijŇæĹŚäžňåőŽäźL'äžĘBatch Sizeäÿž13ïijŇåIJĺè£ŹäÿĂæ■ěäÿ■ïijŇéĂŽè£ĞèőSizeæİěæĺąæŃ§äÿĂäÿłåď ğBatchçŽĎèő■çżČïijŇäżŐèĂŇè¿¿åĹřäžĘBatch sizeäÿž52çŽĎèő■çżČæŢĹæđIJãĂĆ

åIJĺgradient_merge_configsäÿ■ïijŇavgéĂL'éąźçŤĺäžŐæŐğåĹűæćŕåžęçt'ŕèőąçŽĎå¡ćåijŔïijŽå¡Şè True æŮűïijŇäijŽåŕźærŔæňąçŽĎæćŕåžęæśĆåŠŇåźűåĄŽåźşåİĞïijŻåŔ■áźŃåřĘçŻt'æŐěåŕźæćŕåžęæśĆåŠŇïijŇ

```
dist_strategy = fleet.DistributedStrategy()
# ä¡£çŤĺGradient mergeç■ŰçŢěåźűèő¿ç¡őçŽÿåĚşåŔĆæŢř
dist_strategy.gradient_merge = True
dist_strategy.gradient_merge_configs = {"k_steps": 4, "avg": True}
optimizer = fluid.optimizer.Adam(learning_rate=configs.lr)
optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)
```

åijĂåğŃèő■çżČ

Gradient Merge çŽĎèő■çżČäžčçăĄäÿŐ Recompute ç■ŰçŢěçŽÿåŘŇïijŇçŤĺæĹůä¡£çŤĺäÿd'èąŇäžčçăĄå■

```
trainer = X.MultiGPUTrainer()
trainer.fit(model, data_loader, start_step=10)
```

è£ŘèąŇèő■çżČèĎŽælJň

```
fleetrun --gpus 0,1,2,3,4,5,6,7 bert_gradient_merge.py
```

CHAPTER 13

ä¡£çŤÍLARS / LAMB äijŸåŇŰåĹĘåÿČåijŔèűĚåď̆gbatch èő■çżČ

13.1 çőĂäżŃ

åIJÍæŢřæ■őåźűèąŇåĹĘåÿČåijŔèő■çżČåIJžæŹŕäÿ■, åÿÿä¡£çŤĺåćđåŁăGPUæŢřéĞŔçŽĎæŰźåijŔæİěåŁăéäÿžäžĘä£İèŕĄGPUçŽĎçőŮåŁŻå¿ŮåĹřåĚĚåĹĘåĹl'çŤÍ, æŕŔåijăGPUå■ąäÿŁçŽĎbatch sizeéČ¡éIJĂèęĄèűşåď§åď̆ğ. åŻăæ■d'åIJÍåćđåŁăGPU æŢřéĞŔåŘŇæŮű, èő■çżČçŽĎåĚĺåśĂ-batch size äź§äijŽåŔŸåď̆ğ.

ä¡ĘèűŁåď̆ğçŽĎåĚĺåśĂbatch size äijŽåÿęæİěèő■çżČçŽĎæŤűæŢŻéŮőécŸ[1] [2]:

- æĺąåđŃæIJĂçżĹçš¿åžęæ■§åď́ś
- æŤűæŢŻéĂ§åžęåŔŸæĚć, éIJĂèęĄæŻť'åď'ŽçŽĎepoch æĽ'■èČ¡æŤűæŢŻ

LARS[3] åŠŇ LAMB[4] äÿď'äÿłäijŸåŇŰç■ŰçŢěåÿÿçŤĺæİěèğčåĘşäÿŁè£řèűĚåď̆ğbatch èő■çżČäÿ■çŽĎæŤűæŢŻéŮőécŸ.

Paddle åőđçŐřäžĘè£Źäÿďçğ■äijŸåŇŰç■ŰçŢěïijŇpaddle.distributed.fleet ä¡IJäÿžPaddleéĂŽçŤĺçŽĎåĹĘåÿČåijŔèő■çżČAPIæŔŘä¿ŻäžĘçőĂå■ŢæŸŞçŤĺçŽĎæŐěåŔč, çŤĺæĹůåŔłéIJĂèęĄæůžåŁăăĞăèąŇäžčçăĄ åřšåŔŕářEç■ŰçŢěåŁăăĚěåĹřåŐ§æIJĽçŽĎèő■çżČäÿ■ãĂĆ éĂŽè£Ğè£ŹäÿďäÿłäijŸåŇŰç■ŰçŢě, æĹŚäžňåIJÍèűĚåď̆ğbatch åIJžæŹŕäÿ■åőđçŐřäžĘæŻť'å£ńçŽĎæŤűæŢŻéĂ§åžęåŠŇæŮăæ■§çŽĎçš¿åžę, çżŞåŘĹFleet äÿ■åĚűäżŰçŽĎç■ŰçŢě(e.g. AMP) åŔŕäžěæđĄåď̆ğçijl'ç§■çŽĎèő■çżČæŢť'ä¡ŞçŽĎtime2train.

13.1.1 èŕŢéłŇæŢĹædlJ

resnet50 imagenet	Global batch size	epoch	top1
[Goyal et al]	8k	90	76.3%
LARS Paper	32k	90	72.3%
[fleet: lars + amp]	16k	60	76.2%
[fleet: lars + amp]	32k	62	75.9%

13.2 LARS

æĹŚäżňäżěåIJĺå■ŢæIJžåď Žå■ąäÿŁResent50 èő■çżČäÿžçőĂå■Ţä¿Ńå■ŘäżŃçż■fleet äÿ■ LARSçŽĎçŤĺæşŢãĂĆ

```
import os
import fleetx as X
import paddle
paddle.enable_staic()
import paddle.fluid as fluid
import paddle.distributed.fleet.base.role_maker as role_maker
import time
import paddle.distributed.fleet as fleet
```

éĂŽè£ĞX.parse_train_configs()æŐěåŔčïijŇçŤĺæĹůåŔŕäżěåőŽäźL'èő■çżČçŽÿåĚşçŽĎåŔĆæŢ init()æŐěåŔčåőŽäźL'äžĘåĹĘåÿČåijŔæĺąåđŃïijŇäÿŃéÍćäžčçăĄäÿ■çŽĎis_collective=Trueèąĺçđ'ž

```
paddle.enable_static()
configs = X.parse_train_configs()
fleet.init(is_collective=True)
```

çŤĺæĹůåŔŕäżěéĂŽè£ĞX.applicationsæŐěåŔčåŁăè¡¡æĹŚäżňécĎåĚĹåőŽäźL'åě¡çŽĎæĺąåđŃïijŇåę

```
model = X.applications.Resnet50()
downloader = X.utils.Downloader()
local_path = downloader.download_from_bos(
    fs_yaml='https://fleet.bj.bcebos.com/test/loader/small_imagenet.
↪yaml',
    local_path='./data')
batch_size = 32
loader = model.get_train_dataloader(local_path, batch_size=batch_
↪size)
```

LARS äijŸåŇŰçőŮæşŢçŽĎåĚňåijŔåęĆäÿŃ:

$$local\_learning\_rate = learning\_rate * lars\_coeff * \frac{||param||}{||gradient|| + lars\_weight\_decay * ||param||}$$

$$velocity = mu * velocity + local\_learning\_rate * (gradient + lars\_weight\_decay * param + epsilon)$$

$$param = param - velocity$$

åŕŕäžěçIJŃåĹřLARS åĚűåőđæŸŕåIJĺ åÿęweight decay çŽĎmomentum äijŸåŇŰåŹĺçŽĎå§žçąĂäÿŁåŁăåĚěäžĘlocal learning rate çŽĎéĂżè¿Ś, åŕźæŕŔäÿĂåśĆçŽĎlearning rate è£ŻèąŇäžĘæŤ¿çijl'. fleet åřĘ LARSåőđçŐřäÿžäÿĂäÿł fleet meta optimizer, åIJĺä¡£çŤĺæŮűéIJĂèęĄè ő¿ç¡őäÿĂäÿŃåĞăçĆź:

1. LARS meta optimizer çŽĎ inner optimizer å£Ěéąžäÿž momentum, åźűåIJĺ momentum äÿ■åőŽäźL' mu åŠŇlr åŔĆæŢř.
2. åIJĺ fleet dist_strategy åőŽäźL'LARS çL'źæIJL'çŽĎ lars_coeff åŔĆæŢřåŠŇ lars_weight_decay åŔĆæŢř.
   - LARS åůšçżŔåřĘ weight decay åŇĚåŔńè£ŻåĚňåijŔäÿ■, çŤĺæĹůäÿ■éIJĂèęĄåĘ■åIJĺ optimizeräÿ■èő¿ç¡ő regularization.
   - fleet äÿ■è£ŸæŔŘä¿Ż lars_weight_decay è£ĞæžďçŰçŢě, åŕŕäžěéĂŽè£ĞåIJĺexclude_from_weight_decay åŔĆæŢřåŁăåĚěåŕźåžŤlayer çŽĎ name string, èől'è£ŹäÿĂ layer çŽĎåŔĆæŢřäÿ■è£ŻèąŇ lars weight decay. (éĂŽåÿÿæĹŚäżňåřĘ''BN'' åŔĆæŢř åŠŇ FC_bias äžŐlars weight decay äÿ■è£ĞæžĎ')

```
dist_strategy = fleet.DistributedStrategy()

dist_strategy.lars = True
dist_strategy.lars_configs = {
                    "lars_coeff": 0.001,
                    "lars_weight_decay": 0.0005,
                    "exclude_from_weight_decay": ['batch_norm', '.b_
↪0']
                }

optimizer = fluid.optimizer.Momentum(learning_rate=0.01, momentum=0.
↪9)
optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)
```

è£ŹäÿĂéČĺåĹĘåŠŇfleet äÿ■åĚűäžŰäżžåŁąå§žæIJňçŻÿåŘŇ:

```
place = fluid.CUDAPlace(int(os.environ.get('FLAGS_selected_gpus',␣
↪0)))
exe = fluid.Executor(place)
exe.run(fluid.default_startup_program())

for i, data in enumerate(loader()):
    start_time = time.time()
    cost_val = exe.run(model.main_prog,
                       feed=data,
                       fetch_list=[model.loss.name])

    end_time = time.time()
    print(
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
        "worker_index: %d, step%d cost = %f, speed: %f"
        % (fleet.worker_index(), i, cost_val[0], batch_size / (end_
↪time - start_time)))
```

13.2.1 è£ŘèąŇèő■çżČèĎŽæIJň

åőŇæĹŘäÿŁè£řèĎŽæIJňçŽĎçijŰåĘŹåŘŐïijŇæĹŚäžňářšåŔŕäžěä¡£çŤĺäžěäÿŇåŚ¡äžd'äÿĂèąŇåŘŕåŁĺå■Ţ

```
fleetrun --gpus 0,1,2,3,4,5,6,7 --log_dir log example_lars.py
```

13.3 LAMB

æĹŚäžňäžěåIJĺå■ŢæIJžåd'Žå■ąäÿŁBert èő■çżČäÿžçőĂå■Ţä¿Ńå■ŘäżŃçż■fleet äÿ■LAMB çŽĎçŤĺæşŢ.

```
import os
import fleetx as X
import paddle
paddle.enable_staic()
import paddle.fluid as fluid
import paddle.distributed.fleet.base.role_maker as role_maker
import time
import paddle.distributed.fleet as fleet
```

è£ŹäÿĂæ■ěåŠŇäÿŁæŰĞäÿ■çŽĎLARS äÿĂèĞt'ãĂĆ

```
paddle.enable_static()
configs = X.parse_train_configs()
fleet.init(is_collective=True)
```

è£ŹäÿĂæ■ěåŠŇäÿŁæŰĞäÿ■çŽĎLARS äÿĂèĞt'ãĂĆ

```
model = X.applications.Resnet50()
downloader = X.utils.Downloader()
local_path = downloader.download_from_bos(
    fs_yaml='https://fleet.bj.bcebos.com/test/loader/small_imagenet.
↪yaml',
    local_path='./data')
batch_size = 32
loader = model.get_train_dataloader(local_path, batch_size=batch_
↪size)
```

LAMB äijŸåŇŰçőŮæşŢçŽĎåĚňåijŔåęĆäÿŃ:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$r_t = \frac{m_t}{\sqrt{v_t} + \epsilon}$$

$$w_t = w_{t-1} - \eta_t \frac{\|w_{t-1}\|}{\|r_t + \lambda w_{t-1}\|}(r_t + \lambda w_{t-1})$$

åŠŇLARS çśżäijij, LAMB äźš§ŸŕåIJĺåĘĚåśĆäijŸåŇŰåŹĺçŽĎå§žçąĂäÿŁ, åěŮäžĘäÿĂäÿłlocal learning rate çŽĎéĂżè¿Ś, åŕźæŕŔäÿĂåśĆçŽĎlearning rate è£ŻèąŇäžĘæŤ¿çijľ. fleet åřĘ LAMBåőđçŐřäÿžäÿĂäÿł fleet meta optimizer, åIJĺä¡£çŤĺæŮűéIJĂèęĄèő¿ç¡őäÿĂäÿŃåĞăçĆź:

1. LAMB meta optimizer çŽĎ inner optimizer å£Ěéążäÿž adam, åźűåIJĺ adam äÿ■åőŽäźL' å■ęäźăçŐĞlr, äÿĂéŸű moment çŽĎđæŇĞæŢřèąřåĞŔçŐĞbeta1 åŠŇäžŇéŸűmoment çŽĎđæŇĞæŢřèąřåĞŔçŐĞbeta2 åŔĆæŢř.

2. åIJĺ fleet dist_strategy åőŽäźL'LAMB çL'źæIJL'çŽĎ lamb_weight_decay åŔĆæŢř.

   - LAMB åůšçżŔåřĘ weight decay åŇĚåŘńè£ŻåĚňåijŔäÿ■, çŤĺæĹůäÿ■éIJĂèęĄåĘ■åIJĺ optimizeräÿ■èő¿ç¡ő regularization.
   - fleet äÿ■è£ŸæŔŘä¿Ż lamb_weight_decay è£Ğæżd'ç■ŰçŢě, åŔŕäžěéĂŽè£ĞåIJĺexclude_from_weight_decay åŔĆæŢřåŁäåĚěåŕźäžŤlayer çŽĎ name string, èől'è£ŹäÿĂ layer çŽĎåŔĆæŢřäÿ■è£ŻèąŇ lars weight decay. (éĂŽåÿÿæĹŚäžňåřĘ''LN'' äžŐlamb weight decay äÿ■è£Ğæżd')

```
dist_strategy = fleet.DistributedStrategy()

dist_strategy.lamb = True
dist_strategy.lamb_configs = {
                    'lamb_weight_decay': 0.01,
                    'exclude_from_weight_decay': ['layer_norm'],
                }

optimizer = paddle.optimizer.Adam(learning_rate=0.01, beta1=0.9,␣
↪beta2=0.999)
optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)
```

è£ŹäÿĂéČĺåĹĘåŠŇfleet äÿ■åĚűäžŰäżżåŁąå§žæIJňçŻÿåŘŇ:

```
place = fluid.CUDAPlace(int(os.environ.get('FLAGS_selected_gpus',␣
↪0)))
exe = fluid.Executor(place)
exe.run(fluid.default_startup_program())

for i, data in enumerate(loader()):
    start_time = time.time()
    cost_val = exe.run(model.main_prog,
                       feed=data,
                       fetch_list=[model.loss.name])
```

(äÿŃéąçżğçż■)

(çż■äÿŁéąţ)

```
    end_time = time.time()
    print(
        "worker_index: %d, step%d cost = %f, speed: %f"
        % (fleet.worker_index(), i, cost_val[0], batch_size / (end_
↪time - start_time)))
```

13.3.1 è£ŘèąŇèő■çżČèĎŽæIJň

åőŇæĹŘäÿŁè£řèĎŽæIJňçŽĎçijŰåĘŹåŘŐïijŇæĹŚäżňářšåŔŕäžëä¡£çŤĺäžëäÿŃåŚ¡äžď'äÿĂèąŇåŘŕåŁĺå■Ţ

```
fleetrun --gpus 0,1,2,3,4,5,6,7 --log_dir log resnet50_lamb.py
```

CHAPTER 14

ä¡£çŤÍFleetè£Žèą̈ŇåijĆæđĎåŔĆæŢřæIJ■åŁąåŹÍèő■çżČ

åijĆæđĎåŔĆæŢřæIJ■åŁąåŹÍçŻőåL'■äžĚæŤŕæŇĄåIJÍéİŹæĂĄåŻ¿äÿŃè£ŘèąŇ

14.1 äżĂäźĹæŸŕåijĆæđĎåŔĆæŢřæIJ■åŁąåŹÍïij§

åIJÍåijĂåğŃä¡£çŤÍåijĆæđĎåŔĆæŢřæIJ■åŁąåŹÍåL'■ïijŇæĆÍéIJĂèęĄåĚĹäžĘèğčåŔĆæŢřæIJ■åŁąåŹ

14.1.1 åŔĆæŢřæIJ■åŁąåŹÍçŽĎåžŤçŤÍéćĘå§§äżěåŔŁèğčåĘşçŽĎéŮőécŸ

åŔĆæŢřæIJ■åŁąåŹÍéŻĘäÿ■åžŤçŤÍåIJÍNLPãĂĄæŐÍè■Ř äżěåŔŁ
æŘIJçt'ćécĘå§§ïijŇåĚűäÿżèęĄéŠĹårźäżěäÿŃäÿd'äÿłéŮőécŸïijŽ

1. **åd'ğæŢřæ■ő**ïijŽ

   åŐ§åğŃæŢřæ■őéŻĘåžđåd'ğïijŇåŁÍè¿ĎåĞäçŹ¿GçŽĎæŢřæ■őéĞŔïijŇå■ŢæIJżèő■çżČéĂ§åžęéŽ¿äżěæI

2. **åd'ğåŔĆæŢř**ïijŽ

   åIJÍäÿŁè£řåIJżæŹŕäÿ■ïijŇæŢřæ■őéĞŔåd'ğçŽĎåŘŇæŮűïijŇäijt'éŽŔçİĂçL'źå¿ĄçŽĎçÍĂçŰŔæĂğïijŇå

   åIJÍåůěäÿŽåIJżæŹŕäÿ■ïijŇèřěEmbeddingåŔĆæŢřçŽĎçżt'åžęå¿Ăå¿ĂæŸŕäž£çžğïijŇå■ăçŤÍçŽĎå■ŸåĆÍ

14.1.2 äijăçż§åŔĆæŢřæIJ■åŁąåŹÍçŽĎåśĂéŽŘ

å¡ŞåL'■åŔĆæŢřæIJ■åŁąåŹÍçŽĎWorkerèŁĆçĆźïijŇéĂŽåÿÿä¡£çŤÍCPUæĹŰGPUæIJżåŹÍåőŇæĹŘæĺąå

å¡ŞWorkerä¡£çŤÍçŽĎèő¿åd'ĞçąőåőŽïijŇåĚűçąňäżűçőŮåŁŻçŽĎéĚ■æŕŤäź§éŽŘäźŃåŻžåőŽãĂĆåŻžåő

1. **GPUæIJžåŹÍåĹľçŤÍçŐĞè¿Čä¡Ő**

   èŃěèő■çżČçŽĎæÍąåđŃäÿ■åď■æÌĆïijŇåęĆæŐÍè■ŘéćĘå§§åÿÿçŤÍçŽĎDeepFMãĂĄLRïijŇæÍąåđŃèőąç

2. **CPUæIJžåŹÍçőŮåŁŻæIJĽçŞűéćĹ**

   CPUæIJžåŹÍéĂŽåÿÿæăÿå£ČæŢřè¿Čåď'ŽïijŇåźűäÿŤæIJžåŹÍäżůæăijäź§æŻť'ä¿£åőIJïijŇåŕŕäżěåĚĚåĹĘ

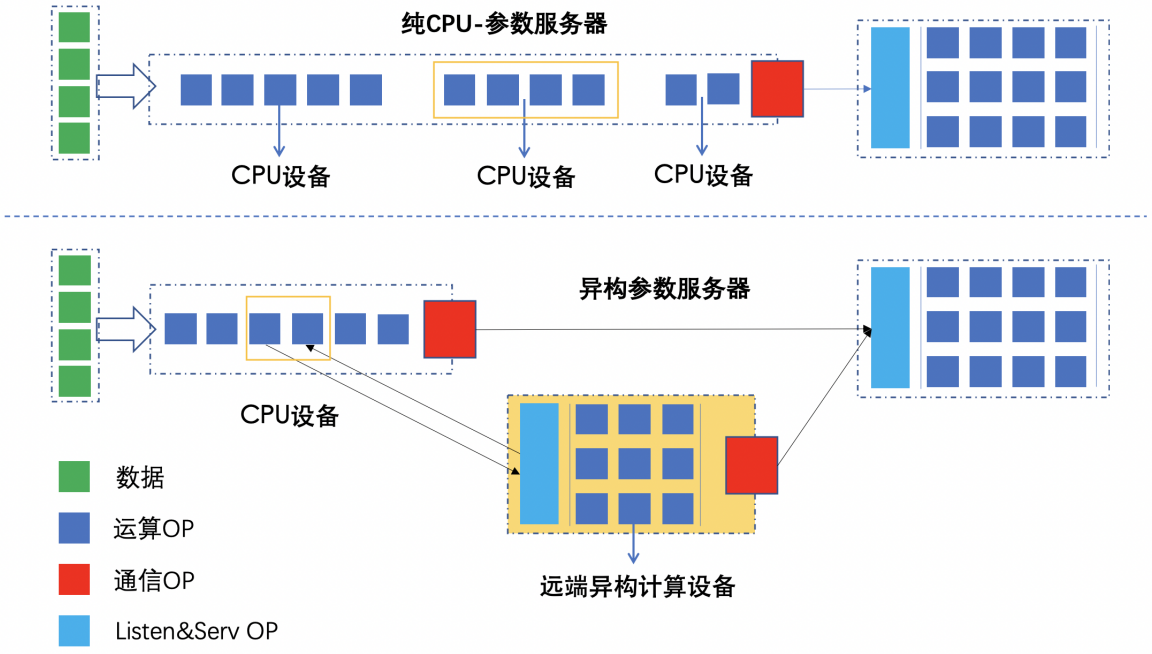
3. **æŰřåđŃçőŮåŁŻæŐěåĚěæĹŘæIJňè¿Čåď'ğ**

   éŽŔçÌĂAIèŁŕçL'ĞåŔŠåśŢæŮěæŰřæIJĹåijĆïijŇåŘĎçğ■énŸçőŮåŁŻä¡ŐæĹŘæIJňçŽĎèŁŕçL'Ğåůšè£Ž

14.1.3 åijĆæđĎåŔĆæŢřæIJ■åŁąåŹÍäżŃçż■

éĆčäźĹïijŇåŕŕäÿ■åŕŕäżěåŁÍæĂĄěřČæŢť'æIJžåŹÍéĚ■æŕŤïij§åŘŇæŮűèğčåĘşIOçŞűéćĹäżěåŔŁçőŮåŁŻç

PaddlePaddleå§žäžŐåůěäÿŽåőđèůţïijŇåĹŻæŰřæĂğçŽĎæŔŘåĞžäžĘåijĆæđĎåŔĆæŢřæIJ■åŁąåŹÍïijŇæ



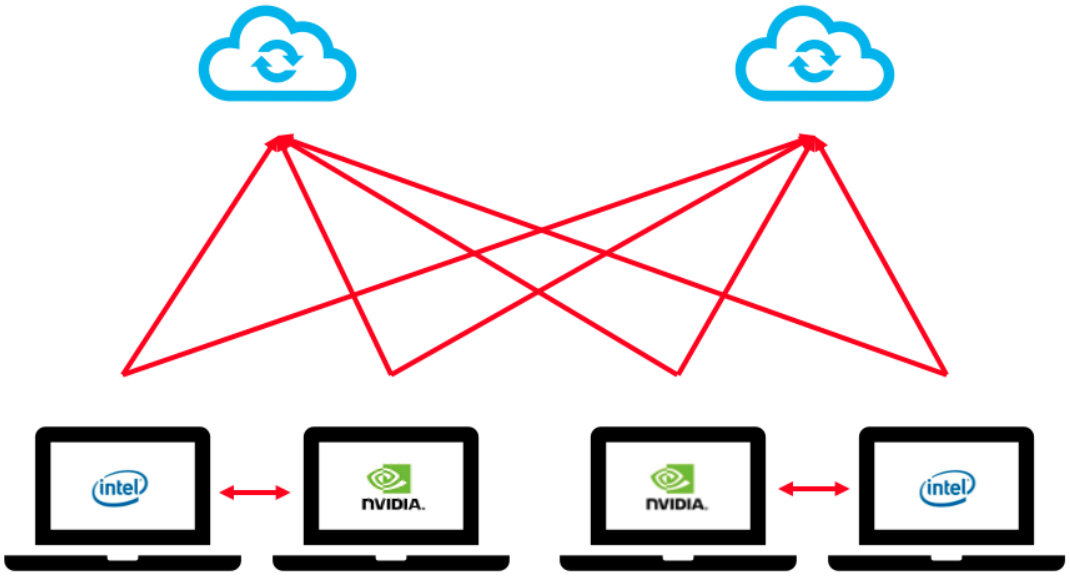
åijĆæđĎåŔĆæŢřæIJ■åŁąåŹÍå§žæIJňåŐ§çŘĘ

äÿĂäÿłæůšåžęå■ęäźăæÍąåđŃçŽĎèő■çżČè£ĞçÍŃåŕŕäżěæŃĘåĹĘäÿžäÿĽ'æ■ëïijŽ1ãĂĄåĽ'■åŘŠèőąçőŮLo

åŔĆæŢřæIJ■åŁąåŹÍæÍąåijŔäÿŃïijŇåĽ'■åŘŠåŔŁåŔ■åŘŠæ■ěéłď'åIJÍWorkerçńŕ(äź§çğřäÿžTrainer)åđ

åijĆæđĎåŔĆæŢřæIJ■åŁąåŹÍæÍąåijŔäÿ■ïijŇæĹŠäžňè£ŽäÿĂæ■ěæŃĘåĹĘåĽ'■åŘŠåŔŁåŔ■åŘŠïijŇåŕŕäż

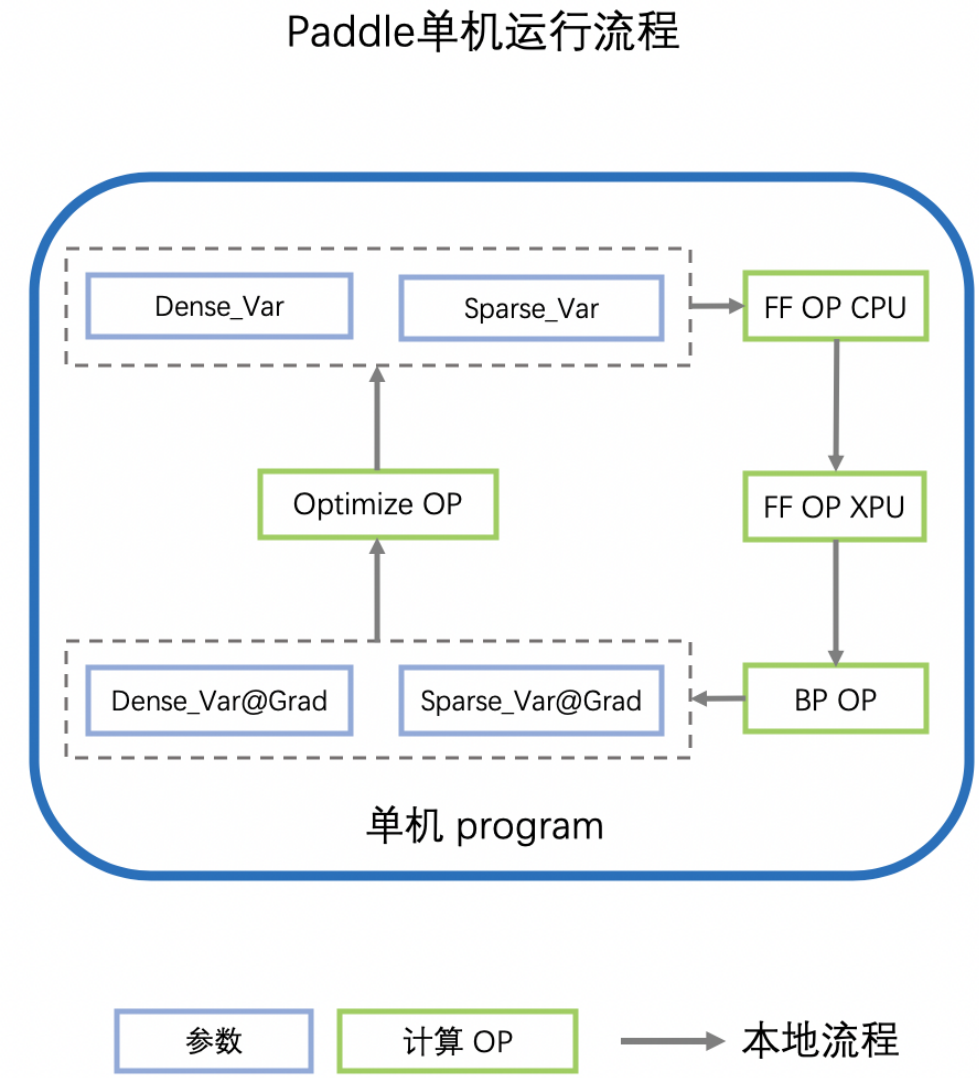
CPU-TraineråŠŇHeter-Traineräź Néůť'äijŽè£ŽèąŇéĂŽä£ąïijŇäžd'æ■éç¡ŠçżIJè£ŘèąŇæĽ'ĂéIJ



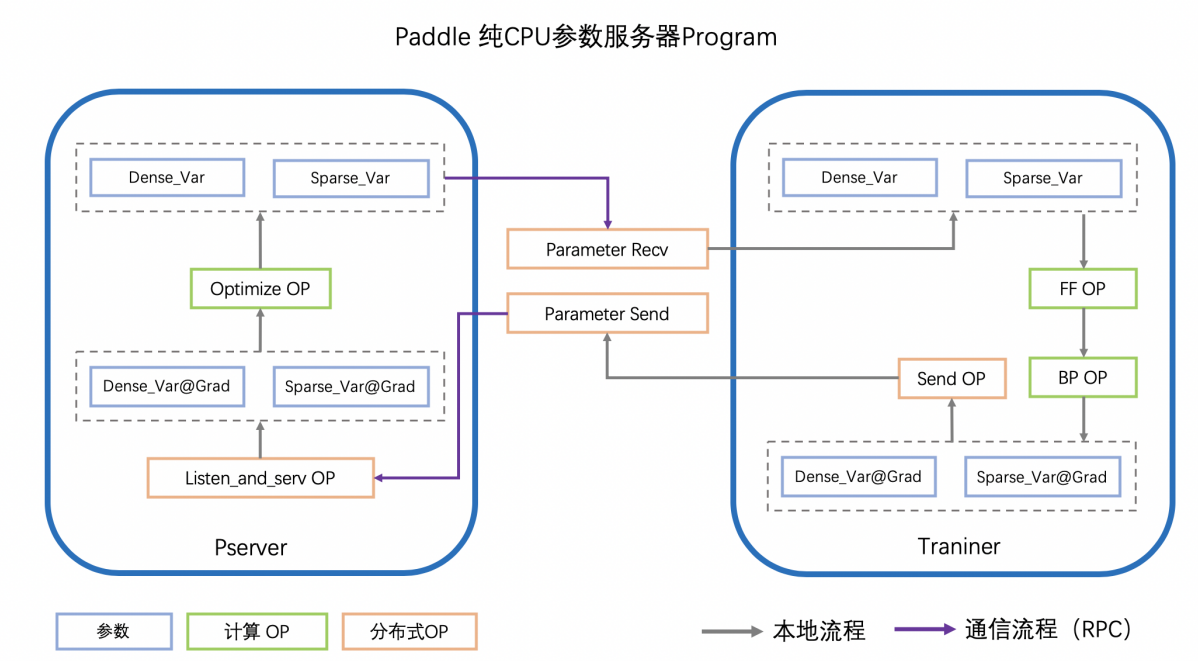
PS-Heter

åijĆædĎåŕĆæŢřælJ■åŁąåŹĺåžŢåśĆåŐ§çŘĘ

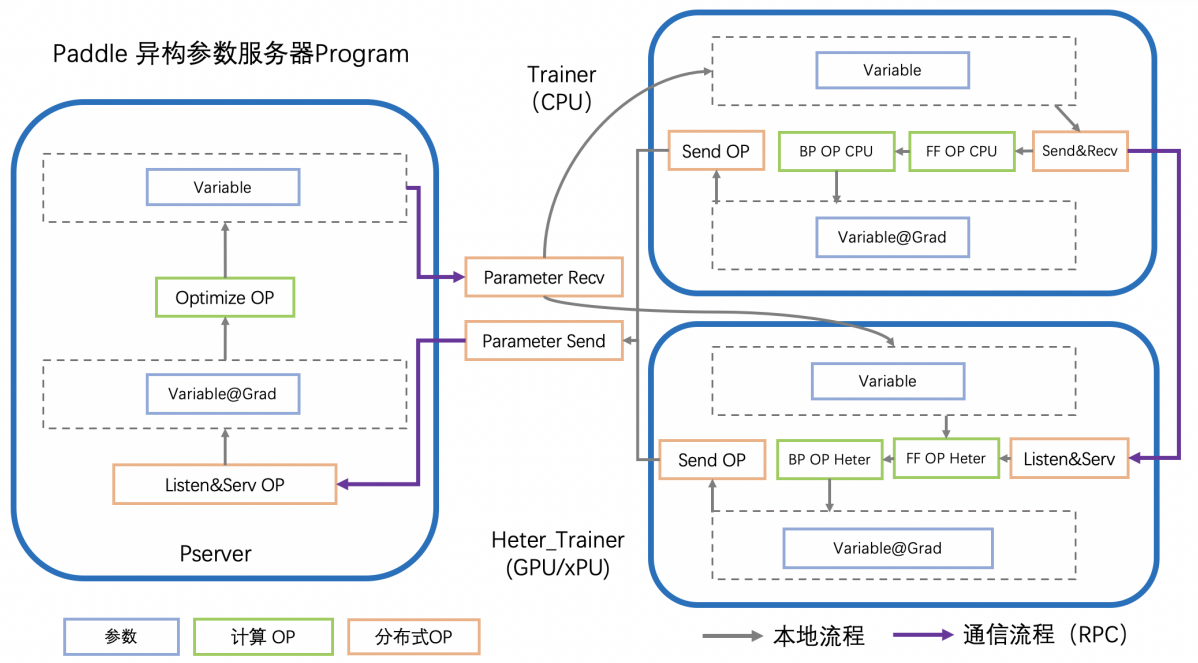
- **å■ŢæIJžèő■çżČçŽĎè£ŘèąŇåŐ§çŘĘåŻ¿**



- **äijăçż§åŔĆæŢřæIJ■åŁąåŹÍçŽĎè£ŘèąŇåŐ§çŘĘåŻ¿**



- **åijĆædĎåŔĆæŢřæIJ■åŁąåŹĺçŽĎè£ŘèąŇåŐ§çŘĘåŻ¿**



åijĆædĎåŔĆæŢřæIJ■åŁąåŹĺä¡£çŤĺæŰźæşŢ

äÿŇéİćäżŇçż■åijĆædĎåŔĆæŢřæIJ■åŁąåŹĺçŽĎä¡£çŤĺæŰźæşŢïijŇæŐĺè■ŘåĚĹåIJĺæ■čåÿÿåŔĆæŢřæIJ■

äžěäÿŇçd'žä¿ŇçŽĎåőŇæŢt'äżčçăĄä¡■äžŐ`FleetX/example/heter_parameter_server/demo.py`

- **1ãĂĄèő¿ç¡őè£ŘèąŇåIJĺåijĆædĎèő¿åd'ĞäÿŁçŽĎçżĎç¡Ś**

æůśåžęå■ęäźăçżĎç¡ŚïijŇéĂŽåÿÿåŔŕäžěæŃĘègčäÿžäÿd'éČĺåĹĘïijŽ1ãĂĄIOåŕĘéŻĘåđŇçżĎç¡ŚïijŻ2ãĂA

```
# --------- IO åŕĘéŽĘåđŃç¡ŚçżIJ ---------
# æŢřæ■őè¿ŞåĚě & embedding æŞěèąÍ & sequence_pool ç■L'æŞ■ä¡IJ
input_data = paddle.data(name="sparse_input", shape=[None, 1],␣
↪dtype="int64")
input_label = paddle.data(name="label", shape=[None, 1], dtype=
↪"int64")
embedding = paddle.static.nn.embedding(input_data, is_sparse=True,␣
↪size=[1000,128])

# --------- èőąçőŮ åŕĘéŽĘåđŃç¡ŚçżIJ ---------
# fc & cnn & rnn & attention ç■L'ç¡ŚçżIJçżŞæđĎ
fc1 = paddle.static.nn.fc(embedding, size=1024, act="relu")
fc2 = paddle.static.nn.fc(fc1, size=512, act="relu")
fc3 = paddle.static.nn.fc(fc2, size=256, act="relu")
predict = paddle.static.nn.fc(fc3, size=2, act="softmax")
cost = paddle.nn.functional.cross_entropy(input=predict,␣
↪label=input_label)
```

æĹŚäžňåŔŕäžëä¡£çŤĺfluid.device_guard()APIåĹŠåĹĘæĺąåđŃäÿ■åŘĎäÿłOPçŽĎè£ŘèąŇèő¿åď

```
with fluid.device_guard("cpu"):
    input_data = paddle.data(name="sparse_input", shape=[None, 1],␣
↪dtype="int64")
    input_label = paddle.data(name="label", shape=[None, 1], dtype=
↪"int64")
    label = paddle.cast(input_label, dtype="float32")
    embedding = paddle.static.nn.embedding(input_data, is_
↪sparse=True, size=[1000,128])


with fluid.device_guard("gpu"):
    fc1 = paddle.static.nn.fc(embedding, size=1024, act="relu")
    fc2 = paddle.static.nn.fc(fc1, size=512, act="relu")
    fc3 = paddle.static.nn.fc(fc2, size=256, act="relu")
    predict = paddle.static.nn.fc(fc3, size=2, act="softmax")
    label = paddle.cast(label, dtype="int64")
    cost = paddle.nn.functional.cross_entropy(input=predict,␣
↪label=label)
```

è£ŹæăůåĹŠåĹĘçżĎç¡ŚçŽĎä¡IJçŤĺæŸŕïijŽ

1. IOåŕĘéŽĘåđŃçŽĎOPéĂĆåŘĹåIJÍCPUèő¿åď ĞäÿŁè£ŘèąŇïijŇä¡£æŢřæ■őè¿ŞåĚèè¿ŞåĞžäÿ■åĘ■æĹŘä
2. èőąçőŮåŕĘéŽĘåđŃOPæŤ¿åIJÍGPUç■L'AIèŁŕçL'Ğèő¿åď ĞäÿŁïijŇåŔŕäžëåĚĚåĹĘåĹl'çŤĺçőŮåŁŻïijŇåŁ

äÿŐæ■ď åŘŇæŮűïijŇPaddle-åijĆæđĎåŘĆæŢřæIJ■åŁąåŹĺïijŇæŤŕæŇĄåźűäÿŤåžžèőőæĆĺåIJÍèő■çżČæŮ TrainerçŽĎèő¿åď ĞæŢřéĞŔ >> Heter-TrainerçŽĎèő¿åď ĞæŢřéĞŔïijŇåŔŕäžëåĚĚåĹĘåćđåď ğæŢřæ■őçŽĎIO

- **2ãĂĄåijĆæđĎåŘĆæŢřæIJ■åŁąåŹĺStrategyéĚ■ç¡ő**

ä¡£çŤĺfleet apiåŘŕåŁĺåijĆæđĎåŘĆæŢřæIJ■åŁąåŹĺïijŇéIJĂèęĄéĚ■ç¡őDistributedStrategyïijŇä¡

```
strategy = paddle.distributed.fleet.DistributedStrategy()
strategy.a_sync = True

optimizer = fluid.optimizer.Adam(args.learning_rate)
optimizer = fleet.distributed_optimizer(optimizer, strategy)
optimizer.minimize(cost)
```

èĂŇåIJÍåijĆæđĎåŔĆæŢřæIJ■åŁąåŹÍæÍąåijŔäÿŃïijŇäżĚéIJĂéćÍåd'ŰæŇĞåőŽåijĆæđĎèő¿åd'Ğä¡£çŤÍçŽ

```
strategy = paddle.distributed.fleet.DistributedStrategy()
strategy.a_sync = True
# ---- æŰřåćđstrategyéĚ■ç¡ő, æŇĞåőŽåijĆæđĎèő¿åd'ĞçŽĎdeviceçśžåđŃ ----
strategy.a_sync_configs = {"heter_worker_device_guard": 'gpu'}

optimizer = paddle.optimizer.Adam(args.learning_rate)
optimizer = fleet.distributed_optimizer(optimizer, strategy)
optimizer.minimize(cost)
```

- **3ãĂĄåijĆæđĎåŔĆæŢřæIJ■åŁąåŹÍçŽĎåŘŕåŁÍçŐŕåćČåŔŸéĞŔéĚ■ç¡ő**

åŘŕåŁÍåijĆæđĎåŔĆæŢřæIJ■åŁąïijŇéIJĂèęĄåIJÍåŔĆæŢřæIJ■åŁąåŹÍçŽĎå§žçąĂäÿŁïijŇäÿžåijĆæđĎèő¿

1. èő¿åd'ĞIPåŔŁéĂŽä£ąçńŕåŘč: `PADDLE_HETER_TRAINER_IP_PORT_LIST=ip:port, ip:port,...`
2. èő■çżČèğŠèL'šçŐŕåćČåŔŸéĞŔ: `TRAINING_ROLE=HETER_TRAINER`

ä¿ŃåęĆïijŽ

```
export PADDLE_HETER_TRAINER_IP_PORT_LIST='ip:port,ip:port'
export TRAINING_ROLE=HETER_TRAINER
```

å¡ŞæL'ğèąŇfleetåĹÍåğŃåŇŰäžčçăĄæŮűïijŽ

```
fleet.init()

# èŃěè£ŽçÍŃæčĂætŃåĹřçŐŕåćČåŔŸéĞŔäÿ■éĚ■ç¡őäžĘ PADDLE_HETER_TRAINER_IP_
↪PORT_
↪LISTïijŇåĹŽäijžè£ŽåĚěåijĆæđĎåŔĆæŢřæIJ■åŁąåŹÍæÍąåijŔïijŇè£ŽèąŇçŽÿåžŤçŽĎèőąçőŮåŽ

# èŃěè£ŽçÍŃæčĂætŃåĹřçŐŕåćČåŔŸéĞŔäÿ■ TRAINING_ROLE␣
↪å■ŸåIJÍïijŇåźűäÿŤç■L'äžŐ HETER_TRAINER␣
↪æŮűïijŇåĹŽèŕěè£ŽçÍŃæL'őæijŤåijĆæđĎèőąçőŮèő¿åd'ĞçŽĎèğŠèL'š

# åijĆæđĎèő¿åd'ĞçŽĎèő¿åd'Ğçśžåđ ŃçŤśäÿŁæŰĞäÿ■æŔŘåĹřçŽĎ strategy.a_
↪sync_configs = {"heter_worker_device_guard": 'gpu'} æŇĞåőŽãĂĆ
```

æĹŚäžňæŔŘä¿ŻäžĘäÿĂéŤőåŘŕåŁÍçŽĎfleetrunåŁ§èČ¡ïijŇåŔŕäžěä¿£åĹl'çŽĎåŘŕåŁÍåijĆæđĎåŔĆæ

ä¡£çŤĺfleetrunåŘŕåŁĺåijĆæđĎåŔĆæŢřæIJ■åŁąåŹĺèő■çżČ

fleetrun æŸŕ paddle2.0rcçĽ'ĹæIJňäžěåŘŐæŰřåŁăåĚěçŽĎåĹĘåÿČåijŔèő■çżČåŘŕåŁĺåůěåĚůïijŇ

å¡Şèő■çżČäžčçăĄreadyäžěåŘŐïijŇåĄĞåęĆèő■çżČåŘŕåŁĺåĚěåŔčæŸŕtrain.pyïijŇåĹŹåŘŕæŇL'çĚğäžěäÿŃçŽĎæŰźåijŔåŘŕåŁĺåijĆæđĎåŔĆæŢřæIJ■åŁąåŹĺèő■çżČïijŽ

æŰźæşŢäÿĂïijŇéŠĹåŕźå■ŢæIJžæĺąæŃ§åĹĘåÿČåijŔèő■çżČïijŇä¡£çŤĺèĞłåŁĺåĹĘéĚ■çŽĎipåŠŇport

```
fleetrun --server_num=2 --worker_num=2 --heter_worker_num=2 train.py
```

æŰźæşŢäžŇïijŇéŠĹåŕźå■ŢæIJžïijŇæĹŰèĞłåőŽäźL'çŽĎåd'ŽæIJžèő■çżČïijŇä¡£çŤĺæŇĞåőŽçŽĎipåŔŁç

```
fleetrun --servers=ip:port,ip:port --workers=ip:port,ip:port --
↪heter_workers=ip:port,ip:port train.py
```

æŰźæşŢäÿL'ïijŇéŠĹåŕźPaddleCloudåźşåŘŕçŽĎcustom-frameworkæĺąåijŔïijŇæŇĞåőŽäżżåŁąçŽĎåŘŕåŁ

PaddleCloudæŸŕçŹ¿åžęåĘĚéČĺçŽĎæůśåžęå■ęäźääżżåŁąåźşåŘŕïijŇæŔŘä¿ŻäžĘä¿£æ■ůçŽĎæŔŘäžd'æţ

```
# heter_workeræŢřéĞŔäijŽæäźæ■őéĚ■ç¡őçŽĎGPUèő¿åd'ĞæŢřéĞŔèĞłåŁĺèřČæŢt'
#␣
↪æůżåŁäèŕěéĚ■ç¡őæŸŕäÿžäžĘæŇĞåőŽfleetrunè£ŘèąŇåIJĺåijĆæđĎåŔĆæŢřæIJ■åŁąåŹĺæĺąåijŔ
fleetrun --heter_worker_num=2 train.py
```

åijĆæđĎåŔĆæŢřæIJ■åŁąåŹĺä¡£çŤĺçd'žä¿Ń

çd'žä¿ŃäžčçăĄä¡■äžŐFleetX/example/heter_parameter_server/

- **æŢřæ■őäÿŃè¡¡**

```
bash sh download_data.sh
```

æL'ğèąŇèŕěèĎŽæIJňïijŇäijŽäžŐåŽ¡åĘĚæžŘçŽĎæIJ■åŁąåŹĺäÿŁäÿŃè¡¡CriteoæŢřæ■őéŻĘïijŇåźűèğčåŐ/train_data_full/ïijŇåĚĺéĞŔæţŃèŕŢæŢřæ■őæŤ¿ç¡őäžŐ./test_data_full/ïijŇçŤĺäžŐå£ńéĂ§élŃèŕĄçŽĎèő■çżČæŢřæ■őäÿŐæţŃèŕŢæŢřæ■őæŤ¿ç¡őäžŐ./train_data/äÿŐ./test_data/ãĂĆ

èĞşæ■d'ïijŇæĹŚäžňåůšåőŇæĹŘæŢřæ■őåĞĘåd'ĞçŽĎåĚĺéČĺåůěä¡IJãĂĆ

- **åŘŕåŁĺèő■çżČ**

```
# ps-cpu
fleetrun --server_num=2 --worker_num=2 heter_train.py

# ps-heter
fleetrun --server_num=2 --worker_num=2 --heter_worker_num=2 heter_
↪train.py
```

CHAPTER 15

1. éčđæąĺåžŢåśĆåĹĘåÿČåijŔAPIçŽĎä¡£çŤĺæąĹä¿Ń

15.1 1.1 çőĂäżŃ

å■ůçğŕçěđçżŔç¡ŚçżIJäÿżèęĄåŇĚåŘńäÿď'çğ■çśžåđŇçŽĎæĺąåđŃåśĆïijŽå■ůçğŕåśĆåŠŇåĚĺè£đæŐěåśĆãĂ
95%çŽĎæĺąåđŃèőąçőŮéĞŔïijŻåĚĺè£đæŐěåśĆåŇĚåŘńçžę95%çŽĎæĺąåđŃåŔĆæŢřéĞŔåŠŇçžę5-
10%çŽĎæĺąåđŃèőąçőŮéĞŔãĂĆéĂŽåÿÿæİěèőšïijŇå■ůçğŕåśĆéĂĆåŘĹéĞĞçŤĺæŢřæ■őåźűèąŇïijŇåŻääÿžå■ů

æIJňæŰĞæąčäżěAlexNetç¡ŚçżIJäÿžä¿ŃäżŃçż■åęĆä¡Ţä¡£çŤĺéčđæąĺçŽĎåžŢåśĆéŻĘåŘĹéĂŽä£ąAPIåőđ

15.2 1.2 æĺąåđŃåźűèąŇåŐ§çŘĘåŠŇåőđçŐř

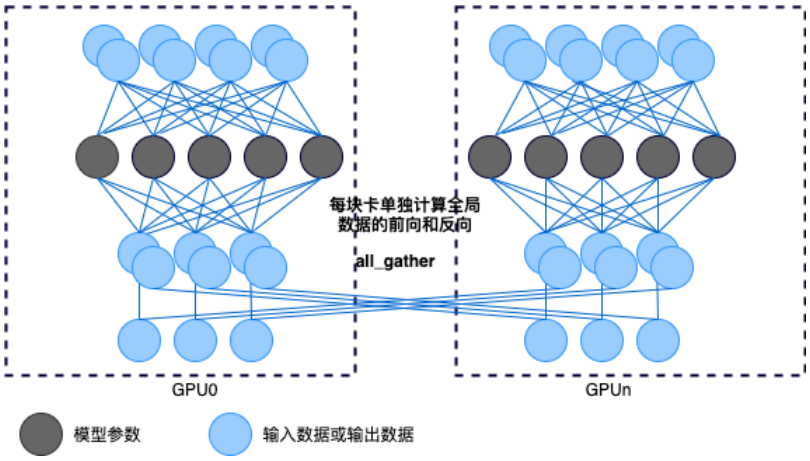
15.2.1 1.2.1 çĽ'ĹæIJňèęĄæśĆ

- paddlepaddle 2.0-rc-gpuçĽ'ĹæIJňåŔŁäżěäÿŁ

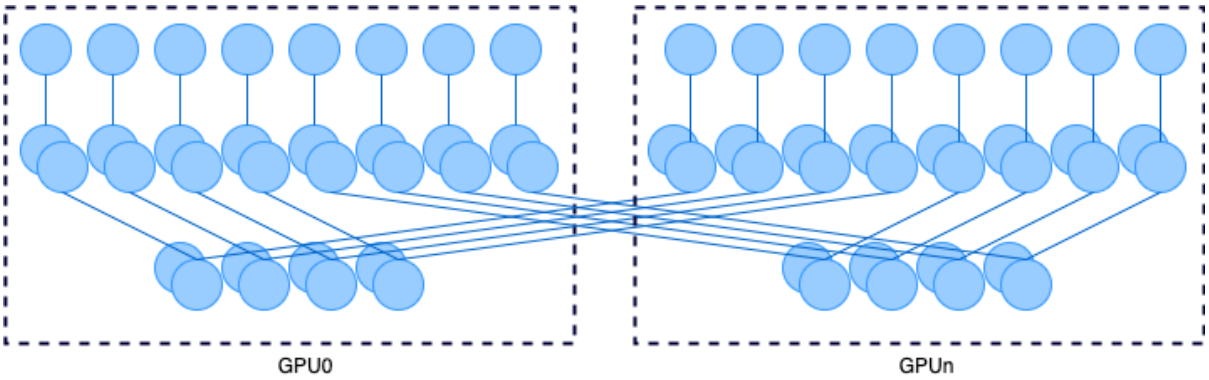
15.2.2 1.2.2 åĹĘåÿČåijŔAPIä¡£çŤĺæąĹä¿Ń

æIJňèŁĆïijŇæĹŚäżňäżŃçż■åęĆä¡ŢåőđçŐřåĚĺè£đæŐěåśĆçŽĎæĺąåđŃåźűèąŇãĂĆ

éęŰåĚĹïijŇæśĞèĄŽåŘĎåİŮGPUå■ąåĚĺè£đæŐěåśĆçŽĎè¿ŞåĚěæŢřæ■őïijŇå¿ŮåĹřåĚĺåśĂè¿ŞåĚěæŢřæ



æŐěçİĂïijŇæśĞèĄŽåŘĎåİŮGPUå■ąåĚĺè£đæŐěåśÇçŽĎè¿ŞåĞžæŢřæ■őïijŇåźűæŁ¡åŔŰæIJňåİŮGPUçŽ



15.2.3 1.2.3 åŁĺæĂĄåŻ¿åŏđçŐř

äÿŁè£řè£ĞçÍŃæŔŔè£řåIJřåŏŇæŢt'åL'■åŘŚèŏąçŐÙè£ĞçÍŃåŏđçŐřäżčçăĄåęĆäÿŃïijŽ

```
# -*- coding: UTF-8 -*-
import paddle
import paddle.nn as nn

# åŏŽäźL'æĺąåđŃåźűèąŇçŽĎåĚĺè£đæŐěç¡ŚçżIJïijŇéIJĂèęĄçżğæL'£èĞłnn.
 ↪Layer
class ModelParallelLinear(nn.Layer):
    def __init__(self,
                 in_dim,
                 rank_num,
                 rank_id,
                 class_num):
        super(ModelParallelLinear, self).__init__()
        if class_num % rank_num:
            raise ValueError("Number of classes must be divisible "
                             "the number of ranks.")
        shard_dims = class_num // rank_num
        self.linear = nn.Linear(in_dim, shard_dims)
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
        self.rank_num = rank_num
        self.rank_id = rank_id
        for parameter in self.linear.parameters():
            parameter.is_distributed = True

    def forward(self, x):
        global_x_list = []
        paddle.distributed.all_gather(global_x_list, x)
        global_x = paddle.concat(global_x_list, axis=0)
        out = self.linear(global_x)
        global_out_list = []
        paddle.distributed.all_gather(global_out_list, out)
        all_outs = paddle.concat(global_out_list, axis=1)
        out = paddle.split(all_outs, self.rank_num)[self.rank_id]
        return out
```

åď ĞæşĺïijŽåŻääÿžæŕŕåİŮGPUå■ąä£Íå■ŸéČĺåĹĘåĚÍè£đæŐěåśĆåŔĆæŢřïijŇäÿŁéİćçŽĎä¿Ńå■Řäÿ■èő¿

åőŇæŢťåIJřèő■çżČäžčçăĄåőđçŐřåęĆäÿŃïijŽ

```
# -*- coding: UTF-8 -*-
import paddle
import paddle.nn as nn
import paddle.nn.functional as F
from paddle.fluid.dygraph import Conv2D
#åĹĘåÿČåijŕstep 1: åŕijåĚěpaddle.distributed.fleetåŇĚ
from paddle.distributed import fleet
from model_parallel_linear import ModelParallelLinear

# åőŽäźL'åĚÍè£đæŐěç¡ŚçżIJïijŇéIJĂçżğæL'£èĞłnn.Layer
class SimpleModelParallelClassifierNet(nn.Layer):
    def __init__(self,
                 class_num,
                 rank_num,
                 rank_id):
        super(SimpleModelParallelClassifierNet, self).__init__()
        self.conv1 = nn.Conv2d(3, 64, kernel_size=11, stride=4,␣
↪padding=2)
        self.max_pool1 = nn.MaxPool2d(kernel_size=3, stride=2)
        self.conv2 = nn.Conv2d(64, 192, kernel_size=5, padding=2)
        self.max_pool2 = nn.MaxPool2d(kernel_size=3, stride=2)
        self.conv3 = nn.Conv2d(192, 384, kernel_size=3)
        self.conv4 = nn.Conv2d(384, 256, kernel_size=3)
        self.conv5 = nn.Conv2d(256, 256, kernel_size=3)
        self.max_pool5 = nn.MaxPool2d(kernel_size=3, stride=2)
        self.model_parallel_linear1 = ModelParallelLinear(2304,
                                                          rank_num,
                                                          rank_id,
                                                          4096)
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
        self.model_parallel_linear2 = ModelParallelLinear(4096,
                                                           rank_num,
                                                           rank_id,
                                                           4096)
        self.model_parallel_linear3 = ModelParallelLinear(4096,
                                                           rank_num,
                                                           rank_id,
                                                           class_num)
        self.droupout = nn.Dropout(0.5)
        self.relu = nn.ReLU()

    def forward(self, x):
        x = self.conv1(x)
        x = self.relu(x)
        x = self.max_pool1(x)
        x = self.conv2(x)
        x = self.relu(x)
        x = self.max_pool2(x)
        x = self.conv3(x)
        x = self.relu(x)
        x = self.conv4(x)
        x = self.relu(x)
        x = self.conv5(x)
        x = self.relu(x)
        x = self.max_pool5(x)
        x = F.dropout(x, 0.5)
        x = paddle.reshape(x, [x.shape[0], -1])
        x = self.model_parallel_linear1(x)
        x = F.dropout(x, 0.5)
        x = self.model_parallel_linear2(x)
        out = self.model_parallel_linear3(x)
        return out

# åĹĘåÿČåijŕstep 2: åĹİåğŃåŇŰfleet
fleet.init(is_collective=True)

# 1. åőŽäźL'ç¡ŚçżIJåŕźèśąïijŇæ■Šåd'śåĞ¡æŢřåŠŇäijŸåŇŰåźÍ
layer = SimpleModelParallelClassifierNet(class_num=1000,
                                         rank_num=fleet.worker_
↪num(),
                                         rank_id=fleet.worker_
↪index())
adam = paddle.optimizer.Adam(learning_rate=0.001,
                             parameters=layer.parameters())

# åĹĘåÿČåijŕstep 3:␣
↪éĂŽè£ĞfleetèŐůåŔŰåĹĘåÿČåijŕäijŸåŇŰåźÍåŠŇåĹĘåÿČåijŕæĺąåđŃ
adam = fleet.distributed_optimizer(adam)
dp_layer = fleet.distributed_model(layer)
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
for step in range(20):
    # 2. æL'ğèąŇåL'■åŘŚç¡ŚçżIJ
    image = paddle.randn([1, 3, 224, 224], 'float32')
    label = paddle.randint(low=0, high=10, shape=[1,1])
    output = dp_layer(image)
    loss = F.softmax_with_cross_entropy(output, label)
    loss = paddle.mean(loss)

    print("step:{}\tloss:{}".format(step, loss.numpy()))

    # 3. æL'ğèąŇåŕ■åŘŚèőąçőŮåŠŇåŕĆæŢřæŻt'æŰř
    # åĹĘåÿČåijŕstep 4:␣
↪åIJĺæL'ğèąŇåŕ■åŘŚïijĹbackwardåĞ¡æŢřïijL'åL'■åŘŐè£ŻèąŇæ■Šåd'śçijl'æŤ¿åŠŇåŕ■åŘŚæ
    loss.backward()

    adam.step()
    adam.clear_grad()
```

åřĘäÿŁè£řäżčçăĄä£Ĺå■Ÿäÿžtrain.pyïijŇåĄĞèő¿èęĄè£ŘèąŇ2å■ąäżżåŁąïijŇéĆčäźĹåŔłéIJĂèęĄåIJĺåŚ¡äż

```
fleetrun --gpus=0,1 tain.py
```

CHAPTER 16

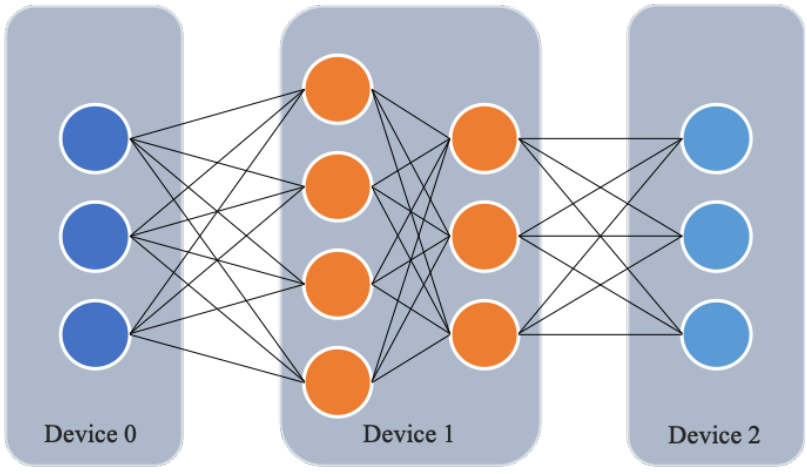
ä¡£çŤĺæţĄæřťçž£åźűèąŇè£Żèą Ňèő■çżČ

16.1 çőĂäżŃ

éŽŔçİĂåď Žçğ■çěđçżŔç¡ŚçżIJåŁăéĂ§èő¿åď ĞåŠŇäÿŞçŤĺçěđçżŔç¡ŚçżIJèőąçőŮèŁŕçL'ĞçŽĎåĞžçŐřïijŇ

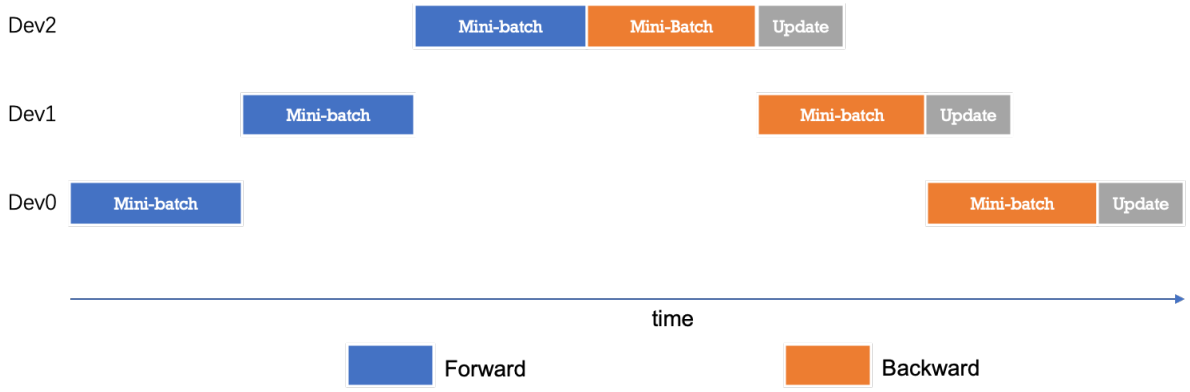
16.2 åŐ§çŘĘ

æţĄæřťçž£åźűèąŇåĹĘåÿČåijŔæŁĂæIJŕäÿŐæŢřæ■őåźűèąŇäÿ■åŘŇïijŇéĂŽè£ĞåřĘæĺąåđŇåĹĞåĹĘåĹř

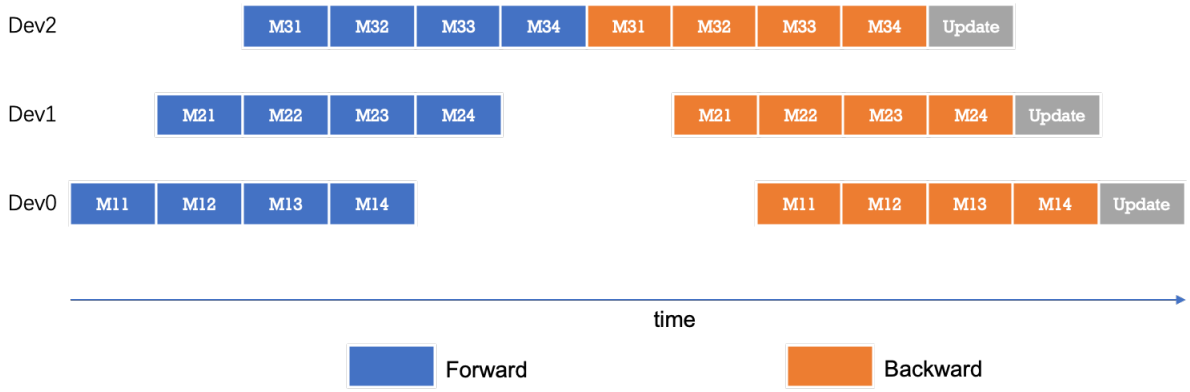


åĚůä¡ŞåIJřèőšïijŇåL'■åŘŚèőąçőŮè£ĞçÍŃäÿ■ïijŇè¿ŞåĚèæŢřæ■őéęŰåĚĹåIJÍèő¿åď Ğ0äÿ■éĂŽè£Ğçňň1å

åęĆäÿŃåŻ¿ïijŇäÿžæţĄæřťçž£åźűèąŇäÿ■çŽĎæŮűåžŔåŻ¿ãĂĆçőĂå■ŢçŽĎæţĄæřťçž£åźűèąŇæŰźåijŔä



äÿžäžĘäijŸåŇŰæţĄæřť çž£åźűèąŇçŽĎæĂğèČ¡ïijŇæĹŚäżňåŕŕäžěåřĘmini-batchåĹĞåĹĘæĹŘèŇěåźšæŻť åřŕçšŠåžęçŽĎmicro-batchïijŇæŔŘå■ĞæţĄæřť çž£åźűèąŇçŽĎåźűåŔŚåžęïijŇèbatchèćńåĹĞåĹĘäÿž4äÿłmicro-batchïijŻåL’■åŘŚéŸűæőţïijŇæŕŔäÿłèő¿åď Ğä¿Ìæňąèőąçő Ůå■Ţäÿłmicro-batchçŽĎçżŞæđIJïijŻè£Źçğ■åĞŔåřŕmini-batchçŽĎæŰźåijŕåĞŔåřŠäžĘæŕŔäÿłèő¿åď ĞåőŇæĹŘäÿĂæňąèőą



äÿŃéÌćæĹŚäżňåřĘéĂŽè£Ğä¿Ńå■Řäÿžæ Ćĺèőšèğčåę Ćä¡Ţä¡£çŤÍpipelineç■ŰçŢěåIJÍäÿd’åijăGPUäÿŁèő■ç

16.3 ä¡£çŤÍæăůä¿Ń

16.3.1 åŕijåĚěä¿ÌètŰ

```
# -*- coding: UTF-8 -*-
import os
import argparse
import paddle
import time
import math
import numpy as np

import paddle.distributed.fleet as fleet
import paddle.static.nn as nn
paddle.enable_static()
```

16.3.2 åőŽäźĽæÍąåđŃ

åIJÍä¡£çŤÍæţĄæřť'çž£åźűèąŇçŽĎèő■çżČç■ŰçŢěæŮűïijŇæĹŚäżňéĂŽè£Ğdevice_guardæŐěåŔčářĘ

åŕźäžŐCPUèő¿åď'ĞïijŇåIJÍä¡£çŤÍdevice_guardæŮűåŔłéIJĂèęĄæŇĞåőŽèő¿åď'ĞçśžåđŃïijŇå■şdev

åIJÍäÿŃéÍćçŽĎä¿Ńå■Řäÿ■ïijŇæĹŚäżňářĘæŢřæ■őåśĆåŔŁembeddingåśĆæŤ¿ç¡őåIJÍCPUäÿ■,
åźűářĘfcåŔŁlossæŤ¿ç¡őåIJÍçňň0åŘůGPUå■ąäÿŁãĂĆ

```
# æÍąåđŃçżĎç¡Ś
def build_network():
    # Step1: ä¡£çŤÍdevice_guradæŇĞåőŽçŽÿåžŤåśĆçŽĎèőąçőŮèő¿åd'Ğ
    with paddle.fluid.device_guard("cpu"):
        data = paddle.data(name='sequence', shape=[1], dtype='int64
↪')
        data_loader = paddle.io.DataLoader.from_generator(
            feed_list=[data],
            capacity=64,
            use_double_buffer=True,
            iterable=False)
        emb = nn.embedding(input=data, size=[128, 64])
    with paddle.fluid.device_guard("gpu:0"):
        fc = nn.fc(emb, size=10)
        loss = paddle.mean(fc)
    return data_loader, loss
```

16.3.3 åőŽäźĽæŢřæ■őéŽĘåŔŁæćŕåžęæŽťæŰřç■ŰçŢě

åőŽäźL'åőŇæÍąåđŃåŘŐïijŇæĹŚäżňåŕŕäžěçżğçż■åőŽäźL'èő■çżČæL'ĂéIJĂèęĄçŽĎæŢřæ■őïijŇäžěåŔŁè

éĂŽè£Ğèő¿åőŽdist_strategy.pipeline äÿžTrueïijŇářĘæţĄæřť'çž£åźűèąŇçŽĎç■ŰçŢěæ£Ăæť'ż

```
fleet.init(is_collective=True)

data_loader, loss = build_network()

dist_strategy = paddle.distributed.fleet.DistributedStrategy()
dist_strategy.pipeline = True
optimizer = paddle.fluid.optimizer.SGDOptimizer(learning_rate=0.1)
optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(loss)

def train_reader():
    for _ in range(100):
        data = np.random.random(size=[32, 1]).astype("int64")
        yield data
```

16.3.4 åijĂåğŃèő■çżČ

```
place = paddle.CPUPlace()
exe = paddle.static.Executor(place)

data_loader.set_sample_generator(train_reader, batch_size=2)

exe.run(paddle.static.default_startup_program())

data_loader.start()
exe.train_from_dataset(paddle.static.default_main_program())
```

CHAPTER 17

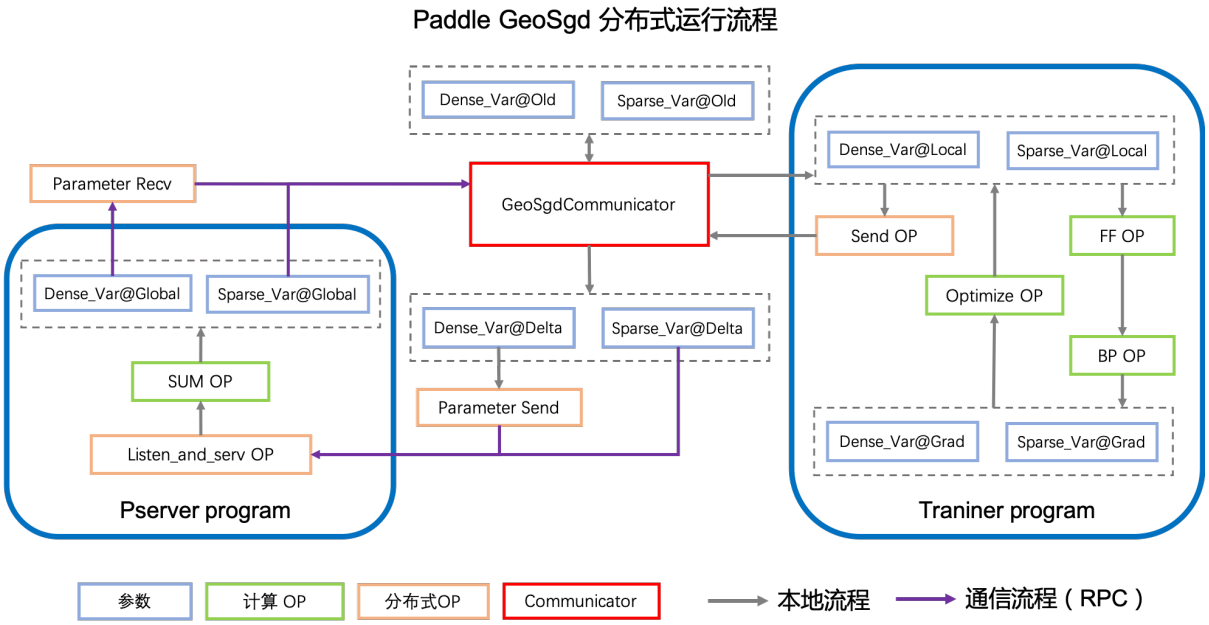
ä¡ŐéćŚéĂŽä£ąåŔĆæŢřæIJ■åŁąåŹĺèő■çżČçőŮæşŢ

17.1 çőĂäżŃ

äijŮæĽĂåŚíç§ěïijŇåIJĺåŘŇæ■ě/åijĆæ■ěåŔĆæŢřæIJ■åŁąåŹĺåĹĘåÿČåijŔèő■çżČäÿ■TraineræŕŔèő■çżČ

äÿžäžĘéŹ■ä¡ŐèŁĆçĆźèğĄéĂŽä£ąåŕźèő■çżČéĂ§åžęçŽĎå¡šåŞ■ïijŇFleetæŘŘä¿ŻäžĘäÿĂçğ■æŻť'éńŸæ

17.2 åŐ§çŘĘ



åIJĺGeoSGDæŻť'æŰřç■ŰçŢěäÿ■ïijŇTrainerçŽĎåŔĆæŢřæŻť'æŰřäź§æŸŕåIJĺåĚĺåijĆæ■ěçŽĎæİąäżűäÿŇ

- äÿŐæŹőéĂŽçŽĎåŔĆæŢřæIJ■åŁąåŹÍäÿ■åŘŇïijŇåIJÍGEOç■ŰçŢěäÿ■ïijŇæŕŔäÿłTrainerèť§èť čåIJÍæIJi
- GEOæŻť æŰřç■ŰçŢěäijŽåIJÍèő■çżČè£ĞçÍŃäÿ■åŘŕåŁÍåď Žäÿłè£ŽçÍŃïijŇèť§èť čåŔĆæŢřæŻť æŰřåŔŁ GEOç■ŰçŢěéĂŽè£ĞæÍąåđŃèő■çżČäÿŐèŁĆçĆźéĂŽä£ąåŘŇæ■ěè£ŻèąŇçŽĎæŰźåijŔïijŇåIJÍä£ÌèŕĄæÍą æŐěäÿŃæÌěæĹŠäžňářĘéĂŽè£Ğä¿Ńå■ŘäÿžæĆÍèőšèğčGEOåIJÍFleetäÿ■æŸŕåęĆä¡ŢåžŤçŤÍçŽĎãĂĆ åIJÍåijĂåğŃäźŃåL'■æĹŠäžńéęŰåĚĹéIJĂèęĄäÿŃè¡¡èő■çżČäÿ■æL'ĂéIJĂèęĄçŽĎæŢřæ■őïijŽ

```
# äÿŃè¡¡åźűèğčåŐŃæŢřæ■őïijŇèő■çżČæŢřæ■őèőšä£Ìå■ŸèĞşåŘ■äÿž raw_data␣
↪çŽĎæŰĞäžűåd'ź
wget --no-check-certificate https://fleet.bj.bcebos.com/ctr_data.
↪tar.gz
tar -zxvf ctr_data.tar.gz
```

17.3 æŞ■ä¡lJåőđèůţ

17.3.1 æůżåŁää¿ìètŰ

éęŰåĚĹæĹŠäžńéIJĂèęĄæůżåŁäèő■çżČäÿ■æL'ĂçŤÍåĹřçŽĎpythonæÍąåİŮïijŇfleetx åŔŕäžěçŤÍäžŐåŁäè¡¡æĹŠäžňäÿžçŤÍæĹůåřĄèčĚçŽĎæŐěåŔčåęĆïijŽåŁäè¡¡æÍąåđŃåŔŁæŢřæ■őïijŇæÍąåđŃèő■ distributed.fleet äÿ■åőŽäźL'äžĘäÿŕårŇçŽĎåĹĘåÿČåijŔç■ŰçŢěä¿ŻçŤÍæĹůä¡£çŤÍãĂĆ

```
import paddle
import fleetx as X
import paddle.fluid as fluid
import paddle.distributed.fleet as fleet
import paddle.distributed.fleet.base.role_maker as role_maker
```

17.3.2 åőŽäźL'åĹĘåÿČåijŔæÍąåijŔåźűåĹìågŃåŇŰ

éĂŽè£ĞX.parse_train_configs() æŐěåŔčïijŇçŤÍæĹůåŔŕäžěåőŽäźL'èő■çżČçŻÿåĚşçŽĎåŔĆæŢ init() æŐěåŔčåőŽäźL'äžĘåĹĘåÿČåijŔæÍąåijŔïijŇåőŽäźL'GEOç■ŰçŢěä¡£çŤÍçŽĎåĹìågŃåŇŰæŐěåŔčäÿĆ

```
paddle.enable_static()
configs = X.parse_train_configs()
role = role_maker.PaddleCloudRoleMaker()
fleet.init(role)
```

17.3.3 åŁäè¡¡æÍąåđŃåŔŁæŢřæ■ő

åIJÍè£Źäÿłä¿Ńå■ŘäÿæĹŠäžňä¡£çŤÍäžĘäÿŐåŘŇæ■ě/åijĆæ■ěåŔĆæŢřæIJ■åŁąåŹÍçŻÿåŘŇçŽĎCTR-DNNæÍąåđŃãĂĆçŤÍX.applicationsæŐěåŔčåŁäè¡¡æÍąåđŃïijŇåźűåŁäè¡¡åőŽåĹűåŇŰçŽĎæŢřæ■őãĂĆ

```
model = X.applications.MultiSlotCTR()
loader = model.load_multislot_from_file('./train_data')
```

17.3.4 åőŽäźĽåŘŇæ■ëèő■çżČ Strategy åŔŁ Optimizer

åIJÍFleet APIäÿ■ïijŇçŤÍæĹůåŔŕäžěä¡£çŤÍfleet.DistributedStrategy()æŐěåŘčåőŽäźL'èĞłåů

æČşèęĄä¡£çŤÍGEOç■ŰçŢěïijŇçŤÍæĹůéęŰåĚĹéIJÄèęĄæL'ŞåijĂåijĆæ■ěåŘĆæŢřæIJ■åŁąåŹÍåijĂåĚşïijŇ TrueãĂĆ

çĎűåŘŐçŤÍæĹůéIJÄèęĄéĂŽè£Ğdistributedstrategy.a_sync_configsèő¿ç¡őTraineräÿŁäijååŔĆæŢ

```
dist_strategy = fleet.DistributedStrategy()
dist_strategy.a_sync = True
dist_strategy.a_sync_configs = {"k_steps": 10000}

optimizer = fluid.optimizer.SGD(learning_rate=0.0001)

optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)
```

17.3.5 åijĂåğŃèő■çżČ

GEOç■ŰçŢěçŽĎèő■çżČäžčçăĄæš£çŤÍäžĘåŔĆæŢřæIJ■åŁąåŹÍåĹĘåÿČåijŔèő■çżČçŽĎå¡ćåijŔãĂĆ

åŕźäžŐServerèŁĆçĆźïijŇéęŰåĚĹçŤÍinit_server()æŐěåŘčåŕźåĚűè£ŻèąŇåĹÌåğŃåŇŰïijŇçĎűåŘŐ

åŘŇæăůåŕźäžŐèő■çżČèŁĆçĆźïijŇçŤÍinit_worker()æŐěåŘčè£ŻèąŇåĹÌåğŃåŇŰåŘŐxïijŇåijĂåğŃ Trainer.fitæŐěåŘčåijĂåğŃèő■çżČãĂĆ

```
if fleet.is_server():
    fleet.init_server()
    fleet.run_server()
else:
    fleet.init_worker()
    trainer = X.Trainer(fluid.CPUPlace())
    trainer.fit(model, loader, epoch=10)
```

17.3.6 è£ŘèąŇèő■çżČèĎŽæIJň

åőŽäźL'åőŇèő■çżČèĎŽæIJňåŘŐïijŇæĹŚäžňåŕśåŘŕäžěçŤÍfleetrunæŇĞäžd'è£ŘèąŇåĹĘåÿČåijŔäżżå worker_numåĹĘåĹńäÿžæIJ■åŁąèŁĆçĆźåŠŇèő■çżČèŁĆçĆźçŽĎæŢřéĞŘãĂĆåIJÍæIJňä¿Ńäÿ■ïijŇæIJ■åŁą

```
fleetrun --server_num=1 --worker_num=2 ctr_app.py
```

CHAPTER 18

æIJ■åŁąåđŃåijźæĂğèŠÿéęŔ

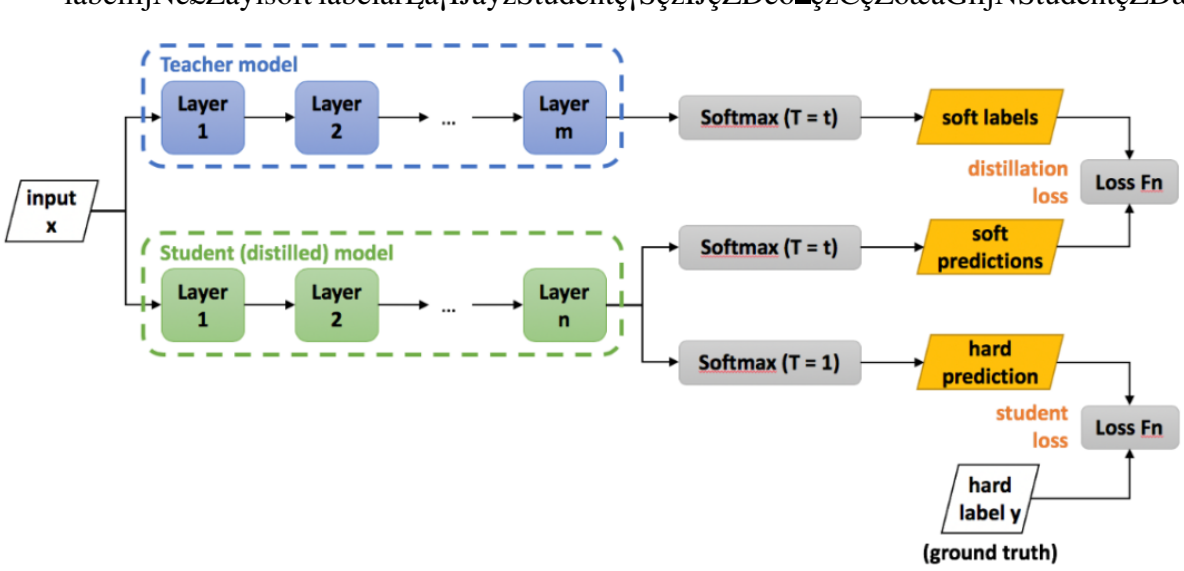
18.1 çőĂäżŃ

18.1.1 èŠÿéęŔèő■çżČ

åIJĺå¿Ĺåď ŽåIJžæŹŕäÿŃïijŇæĺąåđŃèűŁåď ğïijŇåśĆæŢřèűŁåď ŽïijŇæĺąåđŃæŢĹæđIJåřśèűŁåě¡ãĂĆä¡ĘåŔ
the Knowledge in a Neural NetworkãĂŃ åIJĺ2015åźť çňñäÿĂæňąæŔŘåĞžïijŇæŸŕářEç§ěèŕĘäżŐäÿĂäÿłåď ■æ

åęĆäÿŃåŻ¿æĽĂçď žïijŇèő■çżČæ■ěéłď åŔŕäžěåĹĘäÿžäÿď æ■ěïijŽ

- èő■çżČåě¡äÿĂäÿłTeacheræĺąåđŃãĂĆ
- ä¡£çŤĺTeacheræĺąåđŃçŽĎç§ěèŕĘæİěèő■çżČStudentæĺąåđŃãĂĆ
  æĽĂèřŞTeacheræĺąåđŃçŽĎç§ěèŕĘæŸŕæŇĞTeacheræĺąåđŃçŽĎæŐĺçŘĘçżŞæđIJïijŇæĹŚäżňçğřäźŃäÿ
  labelïijŇè£Źäÿłsoft labelåřĘä¡IJäÿžStudentç¡ŚçżIJçŽĎèő■çżČçŻőæăĞïijŇStudentçŽĎæŐĺçŘĘçżŞæđIJ



18.1.2 ælJ■åŁąåđŃèŠÿéęŕèő■çżČ

æIJ■åŁąåđŃèŠÿåŠŇåĚűäżŰåÿÿèğĄèŠÿéęŔæŰźåijŔçŽĎåŕźæŕŤ:

- **çężçž£èŠÿéęŔèő■çżČ**: åĚĹä¡£çŤĺTeacheræĺąåđŃåĄŽæŐĺçŘĘåźűåřĘçżŞæđIJä£Ìå■ŸåIJĺçčĄçŻŸäÿ■ïijI
- **åÿÿèğĎèŠÿéęŔèő■çżČ**: åÿÿèğĎèŠÿéęŔèő■çżČæŸŕæŇĞåřĘ Teacher æĺąåđŃåŠŇ Student æĺąåđŃæŤ¿åĚěåŘŇäÿĂç¡ŚçżIJäÿ■ïijŇåŽžåőŽ Teacher æĺąåđŃåŔĆæŢřåŔłåĄŽåL'■åŘŚïijŇStudent æĺąåđŃåĹŹæ■čåÿÿåĄŽåŔ■åŘŚäijäæŠ■èő■çżČãĂĆè£Źäź§å■Ţè£Źçğ■æŰźåijŔäÿŃ Student æĺąåđŃçŽĎèő■çżČåőŇåĚĺä¿ÌètŰ Teacher æĺąåđŃïijŇStudent æĺąåđŃèęĄç■L' Teacher æĺąåđŃè¿ŞåĞžäÿĂäÿł batch çŽĎæŐĺçŘĘçżŞæđIJæL'■åŔŕäžěèő■çżČïijŇèĂŇ teacher æĺąåđŃäź§èęĄç■L' Student èő■çżČåőŇäÿĂäÿł batchïijŇæL'■èČ¡åijĂåğŃäÿŃäÿĂäÿł batch çŽĎæŐĺçŘĘïijŇåŕźæŢt'ä¡ŞçŽĎèő■çżČéĂ§åžęæIJL'äÿĂåőŽçŽĎå¡śåŞ■ãĂĆ

- **æIJ■åŁąåđŃèŠÿéęŔèő■çżČ**: æŸŕå§žäžŐ Elastic Deep Learning æŔŘåĞžçŽĎäÿĂçğ■èő■çżČæŰźæąĹãĂĆåőČåřĘTeacheræĺąåđŃåŠŇStuden-tæĺąåđŃèğčèĂęïijŇTeacheræĺąåđŃèćńéČĺç¡šäÿžçž£äÿŁæŐĺçŘĘæIJ■åŁąïijŇStudentæĺąåđŃåĹŹäžěåőća

18.1.3 ælJ■åŁąåđŃèŠÿéęŔèő■çżČæŤűçŻŁ

- **èŁĆçžęæŸ¿å■ŸètĎæžŘïijŽ çŤśäžŐStudentæĺąåđŃåŠŇTeacheræĺąåđŃçŽĎèğčèĂęïijŇæL'ĂäžěæIJ■åŁ
- **æŔŘå■Ğèő■çżČéĂ§åžę**ïijŽçŤśäžŐèŁĆçžęäžĘæŸ¿å■ŸètĎæžŘïijŇè£ŹæăůåřśåŔŕäžěä¡£StudentæĺąåđŃsizeïijŻåŘŇæŮűçŤśäžŐStudentæĺąåđŃåŠŇTeacheræĺąåđŃæŸŕåijĆæđĎætĄæřť'çž£ïijŇStudentæĺąåđŃ
- **æŔŘénŸèő■çżČètĎæžŘåĹl'çŤĺçŐĞ**ïijŽæĹŚäžňåŔŕäžěåřĘTeacheræĺąåđŃéČĺç¡šåĹřçž£äÿŁçŽĎåijźæ
- **æŔŘå■Ğèő■çżČæŢĹçŐĞ**ïijŽçŤĺæĹůåŔŕäžěæăźæ■őTeacheråŠŇStudentçŽĎåŘđåŘŘæĂğèČ¡çĄţæť'żè

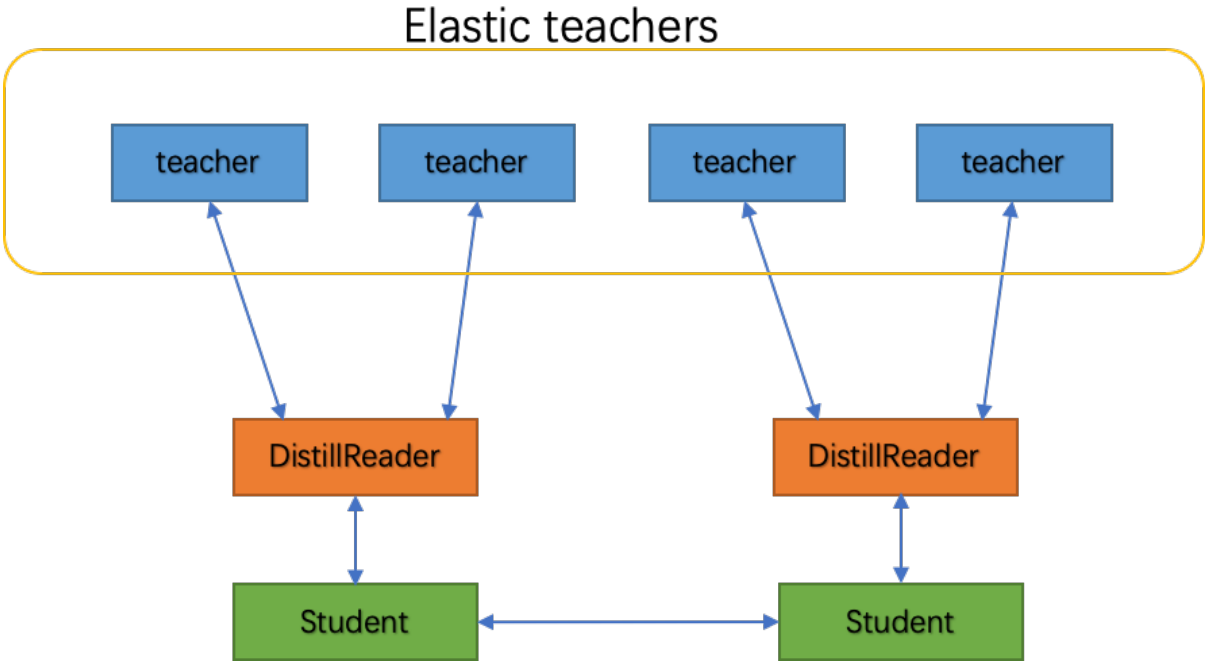
18.2 EDL ælJ■åŁąåđŃåijźæĂğèŠÿéęŔæŢĹæđlJ

ResNet50_vdæĺąåđŃ, ImageNet æŢřæ■őéŻĘ

18.3 ælJ■åŁąåđŃåijźæĂğèŠÿéęŔ

18.3.1 DistillReader

æIJ■åŁąåđŃåijźæĂğèŠÿéęŔçŽĎæăÿå£ĈæŸŕåřĘTeacheræĺąåđŃéČĺç¡šæĹŘäžĘæIJ■åŁąçńŕïijŇèĂŇStudeåřĘTeacheræĺąåđŃèćńéČĺç¡šäÿžåIJĺçž£åŔŕåőźéŤŹåijźæĂğæIJ■åŁą, åIJĺŚ-tudentæĺąåđŃäÿĂä¿ğåĹŹéĂŽè£Ğ `DistillReader` æĺěåřĄèčĚStuden-tæĺąåđŃäÿŐTeacheræĺąåđŃäźŃéŮt'çŽĎéĂŽä£ąïijŇèő£éŮőTeacheræIJ■åŁąĂĆ



DistillReader äžğçŤ§åŕŕä¿ŻStudentæĺąåđŃèő■çżČçŽĎæŢřæ■őreaderãĂĆåęĆäÿŁåŻ¿æL'Ăçd'žïijŇStude
DistillReader çŽĎçżŞæđĎåęĆäÿŃåŻ¿ãĂĆ

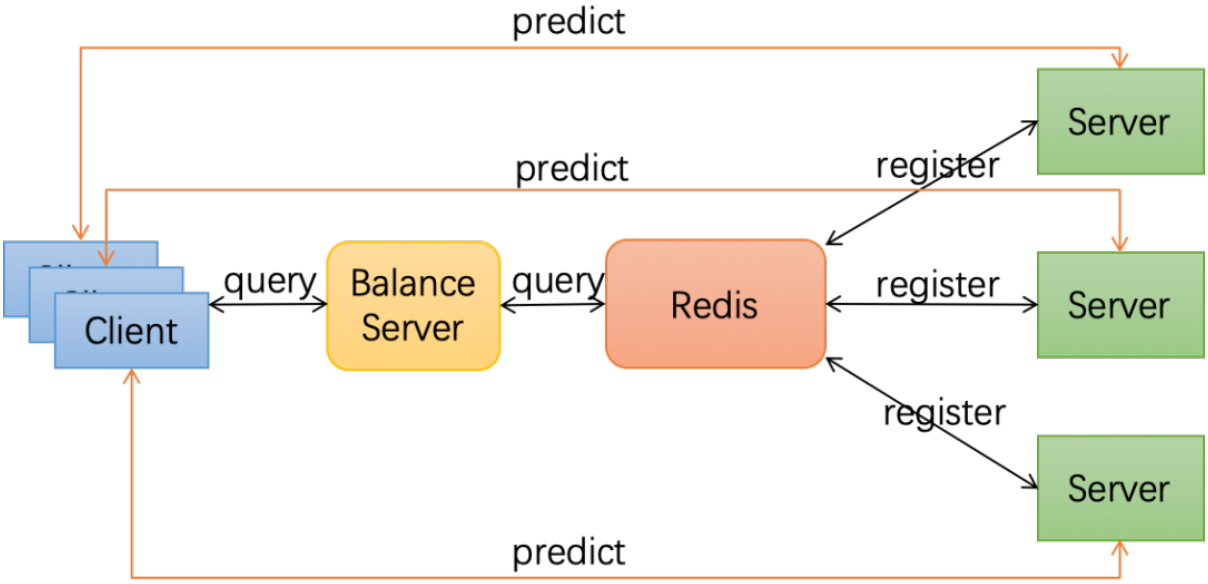
æŐĺçŘĘçżŞæđIJåŠŇåŐ§èő■çżČreaderäÿ■çŽĎæŢřæ■őåřĄèčĚåIJĺäÿĂèţůïijŇè£ŤåŻđäÿĂäÿłåŇĚåŘńæŐ
æĺąåđŃçŽĎæŐĺçŘĘåŠŇStudent æĺąåđŃçŽĎèő■çżČåřśåŔŕäžěæţĄærť'èąŇåźűèąŇèţůæİěäžĘãĂĆ

paddle_fleet_rst/../paddle_fleet/img/distillation_4.png

18.3.2 åŘŕåőźéŤŹåijźæĂğæIJ■åŁą

åŘŕåőźéŤŹåijźæĂğæIJ■åŁąçŽĎåőđçŐřæđűæđĎåęĆäÿŃåŻ¿æL'Ăçd'žïijŇéęŰåĚĹæĹŚäžňéĂŽè£ĞPaddl
ServingåřĘåd'ŽäÿłTeacheræĺąåđŃéČĺç¡šæĹŘæIJ■åŁąïijŇåźűæşĺåĘŇæIJ■åŁąåĹřRedisæŢřæ■őåžŞäÿ■ïijŻStu

Student æĺąåđŃçżŹTeacher æĺąåđŃåŔŚéĂĄæăůæIJňåźűèŐůåŔŰæŐĺçŘĘçżŞæđIJïijŇèĂŇTeacher
æĺąåđŃæIJ■åŁąçńŕåĹŹåŔŕäžěéŽŘæĎŔåćđåĹäïijŇåijźæĂğèřČæŢt'ãĂĆ



18.4 å£ńéĂ§åijĂåğŃ

äÿŃæŰĞéĂŽè£ĞèŏĦçżČåŻ¿åČŔåĹĘçśżæĺąåđŃæİěçőĂå■ŢäžŃçż■æIJ■åŁąåđŃèŠÿéęŔèŏ■çżČçŽĎä¡£ç

äÿžçőĂå■Ţètůèğ ĄïijŇä¡£çŤĺçŽĎæŸŕå■ŢæIJžçŐŕåćČïijŇæIJ■åŁąçńŕåŠŇåőćæĹůçńŕéČĺç¡šåIJĺäžĘåŘŇä

18.4.1 çŐŕåćČåĞĘåd'Ğ

äÿŃåŚ¡äžd'æŃL'åŘŰéŢIJåČŔïijŇéŢIJåČŔäÿžCUDA9.0çŽĎçŐŕåćČïijŇåIJĺéĞŃéİćæĹŚäžňécĎèčĔäžĘĘ
Servingç■L'çŻÿåĚşä¿İètŰãĂĆ

```
docker pull hub.baidubce.com/paddle-edl/paddle_edl:latest-cuda9.0-
↪cudnn7
nvidia-docker run -name paddle_edl hub.baidubce.com/paddle-edl/
↪paddle_edl:latest-cuda9.0-cudnn7 /bin/bash
```

18.4.2 åŘŕåŁÍTeacheræĺąåđŃ

åęĆäÿŃåŚ¡äžd'åIJĺ1åŘůGPUå■qåŘŕåŁÍTeacheræIJ■åŁąijŇåĚűäÿ■TeacheræĺąåđŃäÿžåŻ¿åČŔåĹĘçśżæĺ

```
cd example/distill/resnet

wget --no-check-certificate https://paddle-edl.bj.bcebos.com/
↪distill_teacher_model/ResNeXt101_32x16d_wsl_model.tar.gz
tar -zxf ResNeXt101_32x16d_wsl_model.tar.gz

python -m paddle_serving_server_gpu.serve \
  --model ResNeXt101_32x16d_wsl_model \
  --mem_optim True \
  --port 9898 \
  --gpu_ids 1
```

18.4.3 åŘŕåŁÍStudentæÍąåđŃèő■çżČ

åęĆäÿŃåŚ¡äžď åIJÍ0åŘůGPUå■ąåŘŕåŁÍStudentæÍąåđŃïijŇåŘŕåŁÍçŽĎstudentæÍąåđŃäÿžResNet50_vdãĂ
åĚűäÿ■train_with_fleet.pyæŸŕçŤÍäžŐåŘŕåŁÍèő■çżČçŽĎèĎŽæIJñïijŇçŤÍæĹůéIJĂèęĄåIJÍåĚűäÿ■æůżåŁăèŠÿ

```
python -m paddle.distributed.launch --gpus 0 \
  ./train_with_fleet.py \
  --model=ResNet50_vd \
  --data_dir=./ImageNet \
  --use_distill_service=True \
  --distill_teachers=127.0.0.1:9898
```

18.5 æŐÍè■ŘéŸĚèŕż:

EDLæIJ■åŁąåđŃåijźæĂğèŠÿéęŔGithub éčđæąÍç§ěèŕĘèŠÿéęŔ

CHAPTER 19

äijŸåŇŰä¡ŐéĚ■ç¡ŚçżIJçŽĎåĹĘåÿČåijŔGPUèő■çżČ

åIJÍç¡ŚçżIJåÿęåő¡è¿Čä¡ŐçŽĎèő■çżČåIJžæŹŕïijĺåęĆïijŽ åĚňæIJL'äžŚäÿŁèő■çżČïijŇèĄŤéĆęèő■çżČïijL'
Fleet ä¡IJäÿžPaddleéĂŽçŤĺçŽĎåĹĘåÿČåijŔèő■çżČAPI åőđçŐřäžĘïijŽ `Deep Gradient Compression` åŠŇ `Local SGD` äÿd'çğ■èő■çżČç■ŰçŢěæİěéŠĹářźæĂğèğčåĘşè£ŹäÿĂéŮőécŸãĂĆ

19.1 DGC äijŸåŇŰä¡ŐéĚ■ç¡ŚçżIJçŽĎåĹĘåÿČåijŔGPUèő■çżČ

19.1.1 DGC çőĂäžŃ

åd'ğèğĎæĺąåĹĘåÿČåijŔèő■çżČéIJĂèęĄè¿ČéńŸçŽĎç¡ŚçżIJåÿęåő¡äžëä¿£è£ŻèąŇæćŕåžęçŽĎèĄŽåŘĹæŻ
Deep Gradient Compression åŘŚçŐřïijŽåĹĘåÿČåijŔSGDäÿ■æIJL'99.9%çŽĎæćŕåžęäžd'æ■ćéČ¡æŸŕåĘŮä¡Źç
åőđçŐřäžĘDGCçŽĎçÍĂçŰŔéĂŽä£ąæŰźåijŔïijŇåŘŕæIJL'æŢĹåIJÍä¡ŐéĚ■ç¡ŚçżIJäÿŃè£ŻèąŇGPUåĹĘåÿČåij
åőđçŐřäžĘ DGC èőžæŰĞäÿ■çŽĎ `éćĎçČ■èő■çżČ (warming up training)`, `åŁĺéĞŔä£őæ■č (Momentum Correction)`, `åśĂéČÍæćŕåžęä£őåL'ł (local gradient clipping)`, `åŁĺéĞŔåŻäå■ŘæŐl'èŮŔ (Momentum factor masking)` ç■L'ç■ŰçŢëïijŇ åŠŇ `æ■čåĹŹåŇŰéąźä£őæ■č (Weight Decay Correction)`
éĄ£åĚ■çÍĂçŰŔæćŕåžęéĂŽä£ąèő■çżČåÿęæİęçŽĎæIJĂçżĹæĺąåđŇçš¿åžęæ■§åd'śãĂĆ

äÿŃéİćåřĘäżŃçż■ DGC çÍĂçŰŔéĂŽä£ąæŰźåijŔçŽĎéĂĆçŤĺåIJžæŹŕãĂĄèŕŢéłŇæŢĹæđIJãĂĄå§žæIJňå

éĂĆçŤĺåIJžæŹŕ

DGCçÍĂçŰŔéĂŽä£ąåIJÍä¡Őåÿęåő¡éĂŽä£ąçŞűéĆĹæŮűäijŽæIJL'è¿Čåd'ğçŽĎæĂğèČ¡æŔŘå■ĞïijŇä¡Ę**åI**

19.1.2 èŕŢéłŇæŢĹædlJ

- æĺąåđŃïijŽFasterRCNN
- çąňäžűïijŽ P40äÿď'æIJžåĹĘåÿČåijŔïijŇærŔåŘřæIJžåŹĺäÿĂå■ąïijŇTCPç¡ŚçżIJætŃèŕŢãĂĆ
- åŔŰ300-700æ■ěèĂŮæŮű/400stepãĂĆ
- çš¿åžęæŮăæ■§ãĂĆ

19.1.3 DGC åŐ§çŘĘçőĂäżŃ

è£ŹéĞŇåřĘçőĂå■ŢäżŃçż■äżŃçż■Fleet DGC äÿ■çŽĎäÿĂäžŻåŐ§çŘĘåŠŇåŕźåžŤåŔĆæŢřåžŤèŕěåęĆä¡Ţ

æćŕåžęçÍĂçŰŔ

DGCçŽĎå§žæIJňæĂİèůřæŸŕéĂŽè£ĞåŘłäijăéĂĄéĞ■èęĄæćŕåžęïijŇå■şåŘłåŔŚéĂĄåď'ğäžŐçżŹåőŽéŸÍ
æ■ćäÿłèğŠåžęïijŇäżŐçŘĘèőžä¿İæ■őäÿŁæİěçIJŃïijŇåśĂéČÍæćŕåžęçť'ŕåŁäç■L'åŘŇäžŐéŽŔæŮűéŮť'æŐÍçğžå
sizeïijŇïijÍLDGCçŽÿå¡ŞäžŐærŔäÿĂäÿłæćŕåžęæIJL'èĞłåůśçŽĎbatch sizeïijL'ãĂĆ

åĄĞèő¿ NæŸŕèő■çżČèŁĆçĆźäÿłæŢř, bäÿžå■Ţå■ąbatch
sizeïijŇåśĂéČÍæćŕåžęçť'ŕåŁäåŔŕäžěèćńèőď'äÿžbatch sizeäżŐNbåćđåď'ğäÿžNbTïijŇåĚűäÿ■TæŸŕäÿď'æňąæŻ
[1] éćĎçČ■èřČåŔĆ ^^^^^^^^

åŕźäžŐæ■čåÿÿçŽĎèő■çżČïijŇä¡£çŤÍDGCäÿĂèĹňéIJÀè£ŻèąŇéćĎçČ■èő■çżČïijŇåŘęåĹŹåŔŕèČ¡äijŽæI

```
# 1.␣
↪äžě1252äÿłstepäÿžäÿĂäÿłepochïijŇåL'■2äÿłepochsä¡£çŤÍæ■čåÿÿdenseéĂŽäłąïijŇåŘŐ3ä
↪9%
strategy.dgc_configs = {
    "rampup_begin_step": 1252*2,
    "rampup_step": 1252*3,
    "sparsity": [0.984375, 0.996, 0.999]
}
# 2. åL'■éİć4äÿłepochséČ¡ä¡£çŤÍdenseéĂŽäłąïijŇäźŃåŘŐézŸèőd'0.
↪999çÍĂçŰŔåžęè£ŘèąŇ
strategy.dgc_configs = {
    "rampup_begin_step": 1252*4,
    "rampup_step": 1,
    "sparsity": [0.999]
}
```

åŕźäžŐFine-tuningèő■çżČïijŇåŘŕæŮăéIJĂéćĎçČ■èő■çżČïijŇäżŐçňň0äÿłepochçŽť'æŐëä¡£çŤÍDGCå■şå

```
# äżŐçňň0æ■ěåijĂåğŃDGCçÍĂçŰŔéĂŽäłą
strategy.dgc_configs = {
    "rampup_begin_step": 0,
    "rampup_step": 1,
    "sparsity": [0.999]
}
```

åśĂéČÍæćŕåžęçťŕåŁăæŤźè£Ż

æ■čåÿÿæČĚåĘţïijŇçÍĂçŰŔæŻťæŰřäijŽäÿěéĞ■å¡śåŞ■æŤűæŢŻæĂğãĂĆDGCäÿ■éĞĞçŤÍåŁÍéĞŔä£őæ■
Correction)åŠŇåśĂéČÍæćŕåžęèčĄåĞŔ(Local Gradient Clip-
ping)ïijŇ åŁÍéĞŔåŻăå■ŘæŐľèŮŔïijŇ æ■čåĹŹåŇŰéąźä£őæ■č
4äÿłç■ŰçŢĕæİěèğčåĘşè£ŹäÿłéŮőécŸãĂĆ

åŁÍéĞŔä£őæ■č

äÿŁæŰĞâĂİåśĂéČÍæćŕåžęçťŕåŁăç■L'åŘŇäžŐéŽŔæŮűéŮť'æŐÍçğżåćđåŁăbatch
sizeâĂIJçŽĎæŐÍåŕijæšąæIJL'èĂČèŹŚ Momentumå■ŸåIJÍçŽĎæČĚåĘţãĂĆå¡ŞçÍĂçŰŔåžęå¿ĹénŸæŮűïijŇä¡£
åĚňåijŔäijŽæŸ¿èŚŮéŹ■ä¡ŐæÍąåđŃæĂğèČ¡ïijŇæL'Ăäżěé IJĂèęĄåIJÍåŐ§åğŃåĚňåijŔçŽĎå§žçąĂäÿŁåŕźæćŕå

åŁÍéĞŔä£őæ■čä¡£çŤÍéČÍçť'ŕåŁăéĂ§åžęéąźu_tèĂŇéİđçť'ŕåŁăçIJ§åőđçŽĎæćŕåžę∇_{k,t}æİëä£őæŤźMomentu
æŰźçÍŃïijŇä£őæ■čåŘŐçŽĎåŁÍéĞŔæŻť'æŰŕåĚňåijŔåęĆäÿŃïijŽ

$$u_{k,t} = mu_{k,t-1} + \nabla_{k,t}, \quad v_{k,t} = v_{k,t-1} + u_{k,t}, \quad w_{t+1} = w_t - \eta \sum_{k=1}^{N} \text{sparse}\left(v_{k,t}\right)$$

åśĂéČÍæćŕåžęä£őåL'ł

æćŕåžęä£őåL'łæŸŕéŸšæ■ćæćŕåžęçĹEçĆÿçŽĎåÿÿçŤÍæŰźæşŢãĂĆè£ŹæŰźæşŢçŤśPascanuç■L'äžžåIJÍ20
normsåŠŇåd'ğäžŐçżŹåőŽéŸĹåĂijæŮűïijŇåřśåŕźæćŕåžęrescaleãĂĆæ■čåÿÿæćŕåžęä£őåL'łåIJÍæćŕåžęèĄŽåŘĹ
ãĂĆ

åŁÍéĞŔåŻăå■ŘæŐľèŮŔ

åŽääÿžæŐÍè£§äžĘè¿ČåřŔæćŕåžęæŻť'æŰřæİČéĞ■çŽĎæŮűéŮť'ïijŇæL'Ăäżěäij ŽæIJL'æİČéĞ■éŹĹæŮğæ

$$Mask \leftarrow |v_{k,t}| > thr, \quad v_{k,t} \leftarrow v_{k,t} \odot \neg Mask, \quad u_{k,t} \leftarrow u_{k,t} \odot \neg Mask$$

æ■d'æŐľçăĄåŔŕäžěåĄIJæ■ćåžűè£§æćŕåžęäžğçŤ§çŽĎåŁÍéĞŔïijŇéŸšæ■ćéŹĹæŮğæćŕåžęæŁŁæİČéĞ■åijŢåĚ

æ■čåĹŹåŇŰ(Weight Decay)éąźä£őæ■č

çśżäijijåŁÍéĞŔä£őæ■čïijŇDGC äÿ■æĹŚäžňåŘŇæăůéIJĂèęĄåŕźæ■čåĹŹåŇŰéąźè£ŻèąŇä£őæ■čæİěèőľå

åŠŇåŁÍéĞŔä£őæĂİèůŕçŻÿåŘŇïijŇä£őæ■čéIJĂèęĄåIJÍåśĂéČÍæćŕåžęäÿŁæůżåŁăåśĂéČÍWeight
DecayãĂĆ

$$\nabla_{k,t} = \nabla_{k,t} + \frac{\lambda}{N} w_t$$

äÿŁè£řç■ŰçŢĕåůšçżŔåIJÍFleet æąĘæđűäÿ■åőđçŐřïijŇçŤÍæĹůæŮăéążèő¿ç¡őãĂĆ

19.1.4 DGC å£ńéĂ§åijĂåğŃ

äÿŃæŰĞäżěå■ŢæIJžåĚńå■ąäÿŁèő■çżČResNet50 äÿžä¿Ńå■ŘçőĂå■ŢäżŃçż■ Fleet äÿ■ DGC çŽĎä¡£çŤĺãĂĆ åŻääÿž 8åijă GPU çŽĎéĂŽä£ąéČ¡åIJĺåŘŇäÿĂèŁĆçĆźåĘĚïijŇ äÿĂèĹňæČĚåĘţäÿŃæćŕåžęéĂŽä£ąåźűäÿ■äijŽæĹŘäÿžèő■çżČçŽĎçŞűéĆĹïijŇ è£ŹéĞŇåŘłæŸŕäżěåĚűäÿžä¿Ńå■ŘïijŇäżŃçż■Fleet äÿ■ DGC åŔĆæŢřçŽĎèő¿ç¡õãĂĆ

æşĺæĎŔïijŽ

- çąňäżűçŐŕåćČèęĄæśĆïijŽ DGCçŻőåL'■åŘłæŤŕæŇĄGPUåď'Žå■ąåŘŁåĹĘåÿČåijŔcollectiveèő■çżČïijŇéIJĂèęĄæIJL'çŽÿåžŢçŽĎcudaãĂĄcuDNNãĂĄncclçŐŕåćČãĂĆ
- PaddleçŐŕåćČèęĄæśĆïijŽ DGCåŘłæŤŕæŇĄGPUïijŇæL'Ăäžěéİ JĂGPUçL'ĹæIJňçŽĎ-PaddleãĂĆ

æůżåŁää¿ièţŰ

```
import os
import fleetx as X
import paddle
import paddle.fluid as fluid
import paddle.distributed.fleet.base.role_maker as role_maker
import time
import paddle.distributed.fleet as fleet
```

åőŽäźL'åĹĘåÿČåijŔæĺąåijŔåźűåĹìåğŃåŇŰ

éĂŽè£ĞX.parse_train_configs()æŐěåŔčïijŇçŢĺæĹůåŔŕäžěåőŽäźL'èő■çżČçŽÿåĚşçŽĎåŔĆæŢ init()æŐěåŔčåőŽäźL'äžĘåĹĘåÿČåijŔæĺąåđŃïijŇäÿŃéİćäžčçăĄäÿ■çŽĎis_collective=Trueèąĺçd'ž

```
paddle.enable_static()
configs = X.parse_train_configs()
fleet.init(is_collective=True)
```

åŁăè¡¡æĺąåđŃåŔŁæŢřæ■ő

çŢĺæĹůåŔŕäžěéĂŽè£ĞX.applicationsæŐěåŔčåŁăè¡¡æĹŚäžňéćĎåĚĹåőŽäźL'åě¡çŽĎæĺąåđŃïijŇåę

```
model = X.applications.Resnet50()
downloader = X.utils.Downloader()
local_path = downloader.download_from_bos(
    fs_yaml='https://fleet.bj.bcebos.com/test/loader/small_imagenet.
↪yaml',
    local_path='./data')
batch_size = 32
loader = model.get_train_dataloader(local_path, batch_size=batch_
↪size)
```

DGC çŻÿåĚşç■ŰçŢě

è£ŹéĞŇåĄĞèő¿ïijŽ1252äÿłstepäÿžäÿĂäÿłepochïijŇåĽ■2äÿłepochsä¡£çŤĺæ■čåÿÿdenseéĂŽä£ąïijŇåŘŐ

- `rampup_begin_step (int)` ïijŽDGC(åŘńéćĎçČ■èő■çżČ)åijĂåğŃçŽĎ step
- `rampup_step (int)` ïijŽDGCäÿ■éćĎçČ■èő■çżČæŇĄçż■çŽĎ step. åęĆædIJsparsity æŸŕ [0.75, 0.9375, 0.984375, 0.996, 0.999]ïijŇrampup_step èő¿æĹŘ 100æŮűïijŇåIJĺ 0~19 steps æŮű sparsity=0.75ïijŇåIJĺ 20~39 steps æŮű sparsity=0.9375ïijŇäžěæ■d'çśżæŐĺãĂĆ
- `sparsity (list[float])` ïijŽçĺĂçŰŕåžę threshold, (1 - current sparsity) % çŽĎgradient åřĘäijŽèćń allreduceãĂĆ

```
dist_strategy = fleet.DistributedStrategy()

dist_strategy.dgc = True
dist_strategy.dgc_configs = {
    "rampup_begin_step": 1252*2,
    "rampup_step": 1252*3,
    "sparsity": [0.984375, 0.996, 0.999]
}

optimizer = fluid.optimizer.Momentum(learning_rate=0.01, momentum=0.
↪9)
optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)
```

åijĂåğŃèő■çżČ

è£ŹäÿĂéČĺåĹĘåŠŇFleet äÿ■åĚűäžŰäżżåŁąå§žæIJňçŻÿåŘŇ:

```
place = fluid.CUDAPlace(int(os.environ.get('FLAGS_selected_gpus',␣
↪0)))
exe = fluid.Executor(place)
exe.run(fluid.default_startup_program())

for i, data in enumerate(loader()):
    start_time = time.time()
    cost_val = exe.run(model.main_prog,
                       feed=data,
                       fetch_list=[model.loss.name])

    end_time = time.time()
    print(
        "worker_index: %d, step%d cost = %f, speed: %f"
        % (fleet.worker_index(), i, cost_val[0], batch_size / (end_
↪time - start_time)))
```

è£Řèą Ňèő■çżČèĎŽælJň

äÿĂèąŇåŘŕåŁĺå■ŢæIJžåď Žå■ąåĹĘåÿČåijŕèő■çżČïijŽ

```
fleetrun --gpus 0,1,2,3,4,5,6,7 --log_dir log ./example_dgc.py

# reader shuffle seed 0
# trainerid, trainer_count 0 8
# read images from 0, length: 160146, lines length: 160146, total:␣
↪1281168
# worker_index: 0, step0 cost = 7.151402, speed: 37.698432
# worker_index: 0, step1 cost = 7.112389, speed: 101.518513
# worker_index: 0, step2 cost = 7.004275, speed: 111.062341
# worker_index: 0, step3 cost = 7.039385, speed: 62.173126
# worker_index: 0, step4 cost = 6.985911, speed: 104.058060
# ......
```

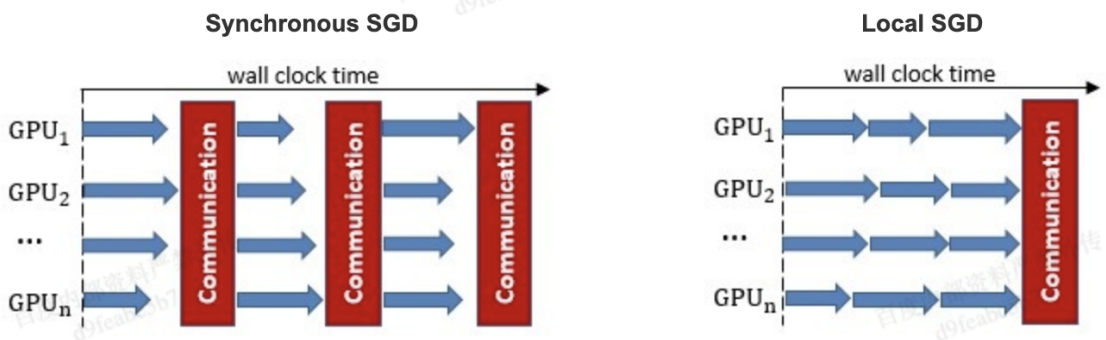
19.2 ä¡£çŤÍLocal SGD äijŸåŇŰä¡Őåÿęåő¡äÿŃåĹĘåÿČåijŕèő■çżČ

19.2.1 Local SGD çőĂäżŃ

åIJÍä¡£çŤÍ distributed SGD è£ŻèąŇæŢřæ■őåźűèąŇçŽĎåĹĘåÿČåijŕèő■çżČæŮűïijŇåÿÿäijŽéĄĞåĹřäžěäÿ

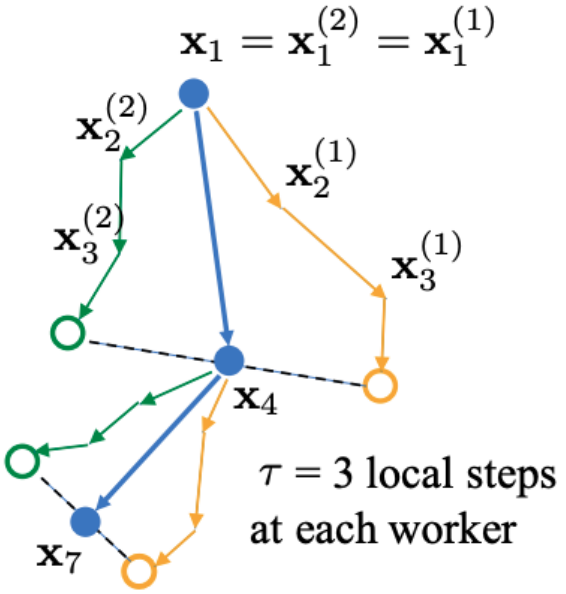
- åĹĘåÿČåijŕèő■çżČçŽĎåŘđåŘŘäijŽåŔŮåĹřéŻĘç¿d'äÿ■éŽŕæIJžæĚćèŁĆçĆźïijĹstraggling nodeïijL'åŠŇéĂŽä£ąåžűê£§çŽĎå¡śåŞ■ãĂĆ
- æŢřæ■őåźűèąŇåĹĘåÿČåijŕåćđåď ğäžĘèő■çżČåőđéŹĚçŽĎbatch sizeïijŇè£Ğåď ğçŽĎbatch size äijŽå¡śåŞ■æIJĂçżĹçŽĎèő■çżČçš¿åžęãĂĆ

Local SGD éĂŽè£Ğåžűéţ£èŁĆçĆźéŮť åŘŇæ■ěçŽĎéŮť éŽŤ(åśÄéČĺåijĆæ■ěèő■çżČ)æİěåĞŕè¡żæĚćèŁ sizeïijL'çŽĎçš¿åžęæ■§åď śïijŇ[1] åŠŇ [2] åĹĘåĹńæŔŘåĞžäžĘïijŽpost-Local SGD åŠŇ èĞłéĂĆåžŤæ■ééŢ£ (Adaptive Communication) Local SGD ç■ŰçŢěïijŇæİěåĞŕåŕŠåŔĆæŢřåŘŇæ■ěécŚçŐĞéŹ■ä¡ŐåÿęæİěçŽĎçš¿åžęæ■§åď śãĂĆ åŘŇæ■ěSGD åŠŇ Local SGD åIJÍéĂŽä£ąåŘŇæ■ěäÿŁçŽĎåůőåijĆåęĆäÿŃåŻ¿æL'Äçď žãĂĆ



åIJÍLocal SGD èő■çżČäÿ■ïijŇéŻĘç¿d'äÿ■çŽĎæŕŔäÿł trainer åŘĎèĞłäijŽçŃňçńNçŽĎè£ŻèąŇ H äÿłè£đçż■çŽĎ SGD æŻť'æŰřïijŇ

çĎűåŘŐéŻEç¿d'äÿ■çŽĎæL'ĂæIJL' trainer äijŽè£Żèą ŇéĂŽä£ąïijŇåŘŇæ■ěïijÍ averagingïijL'æL'ĂæIJL' trainers äÿŁçŽĎåŔĆæŢřãĂĆäÿĂäÿłåŔŇ trainersïijŇåŘŇæ■ěéŮť éŽŹäÿž3 æ■ěéŢ£ïijÍiterationsïijL' çŽĎLocal SGDè£ĞçÍŇåęĆäÿŇåŻ¿æL'Ăçd'žãĂĆéžĎçż£äÿd'æİąèůŕå¿Ďèą Íçd'žäÿd'ä trainers åŘĎèĞłçŽĎ Local SGD æŻť'æŰřè£ĞçÍŃïijŇäÿ■éŮť'çŽĎèŞİèL'šèůŕå¿Ďèą Íçd'žåŘŇæ■ěåŘŐçŽĎæÍąå



Local SGDäÿ■çŽĎäÿĂäÿłåĚşéŤőéŮőécŸæŸŕåęĆä¡ŢçąőåőŽåŔĆæŢřåŘŇæ■ěçŽĎéŮť'éŽŤ(écŚçŐĞ)ïijŇ

- åćđåď'ğåŔĆæŢřåŘŇæ■ěçŽĎéŮť'éŽŤåŔŕäžěåĞŔářŚ trainers éŮť'éĂŽä£ąåžűè£§çŽĎå¡śåŞ■æŔŘéńŸèő■çżĆåŘđåŘŘ.
- ä¡Ęåćđåď'ğåŘŇæ■ěéŮť'éŽŤåŔŕèČ¡äijŽéĂăæĹŘæIJĂçżĹèő■çżĆçš¿åžęçŽĎæ■§åď'śãĂĆ [1]

äžěäÿŃäÿd'äÿłç■ŰçŢěäżŐäÿ■åŘŇèğŠåžęèŕŢåŻ¿è¿¿åĹřæŻť'åě¡çŽĎåźşèąąïijŽ

- post Local SGD åřĘèő■çżĆè£ĞçÍŃåĹĘæĹŘäÿd'äÿłéŸűæőţïijŽçňňäÿĂéŸűæőţ wokers éŮť'åŘŇæ■ěçŽĎéŮť'éŽŤäÿž 1 äÿłæ■ěéŢ£ïijŇå■şåŘŇæ■ěSGDïijŇæİěä£İèŕĄæIJĂçżĹèő■çżĆçš¿åžęïijŽ HïijŇæİěæŔŘå■Ğèő■çżĆåŘđåŘŘãĂĆ
- Adaptive Communication Local SGD éĂŽè£ĞåŁÍæĂĄçŽĎèřČæŢť'åŔĆæŢřåŘŇæ■ěçŽĎéŮť'éŽŤæİěåř

Fleet äÿ■åőđçŐřäžĘ `post Local SGD` åŠŇ `Adaptive Communication Local SGD` äÿd'çğ■ç■ŰçŢěãĂĆ äÿ■äÿŃæŰĞåřĘçżŹåĞž Fleetäÿ■ Local SGD çŽĎåőđèůţæŢĹæđIJïijŇåźűéĂŽè£ĞäÿĂäÿłçőĂå■Ţä¿Ńå■ŘäżNçż■åęĆä¡ŢåIJÍFleet äÿ■ä¡£çŤÍ Local SGDãĂĆ

19.2.2 èŕŢéłŇæŢĹæđIJ

èŕŢéłŇèő¿ç¡ő

model	dataset	local batch size	cluster	dtype	warming up	learning rate decay
resnet50	Imagenet	128	4 x 8 x V100	FP32	30	polynomial

èŕŢéłŇçżŞædIJ

local step	qps	acc1	acc5
1	8270.91	0.7579	0.9266
2	8715.67	0.7533	0.9265
4	8762.66	0.7551	0.9260
8	9184.62	0.7511	0.9239
16	9431.46	0.7429	0.9206
ADACOMM	8945.74	0.7555	0.9270

åŔŕäżěçIJŃåĹřåIJÍ post Local SGD ïijĹåŻžåőŽåŘŇæ■ěéŮťéŽŤïijL'æČĚåĘţäÿŃïijŇæŻť'æŰřéŮťéŽŤèűŁ
å¡Şä¡£çŤÍ ADAPTIVE COMMUNICATION ç■ŰçŢěåŘŐïijŇèő■çżČåIJĺåŘđåŘŘåŠŇçš¿åžęéŮť'è¿¿åĹřäžĘäÿ

19.2.3 Local SGD å£ńéĂ§åijĂåğŃ

äÿŃæŰĞåřĘäżěåIJĺå■ŢæIJž8å■ąäÿ■èő■çżČ ResNet50 äÿžä¿Ńå■ŘçőĂå■ŢäżŃçż■ Fleet äÿ■ Local SGD çŽĎçŤÍæşŢãĂĆ éIJĂèęĄæşĺæĎŔçŽĎæŸŕ å■ŢæIJžåĚńå■ąçŽĎéĂŽä£ąéČ¡åIJĺåŘŇäÿĂèŁĆçĆźåĘĚïijŇ äÿĂèĹňæČĚåĘţäÿŃåŔĆæŢřåŘŇæ■ěåźűäÿ■äijŽ è£ŹéĞŇåŔłæŸŕäżěåĚűäÿžä¿Ńå■ŘïijŇäżŃçż■Fleet äÿ■ Local SGD åŔĆæŢřçŽĎèő¿ç¡őãĂĆ

æůżåŁää¿ÌèţŰ

```
import os
import fleetx as X
import paddle
import paddle.fluid as fluid
import paddle.distributed.fleet.base.role_maker as role_maker
import time
import paddle.distributed.fleet as fleet
```

åőŽäźĽåĹĘåÿČåijŔæĺąåijŔåźűåĹÌåğŃåŇŰ

éĂŽè£ĞX.parse_train_configs()æŐěåŘčïijŇçŤÍæĹůåŔŕäżěåőŽäźĽèő■çżČçŽÿåĚşçŽĎåŔĆæŢ
init()æŐěåŘčåőŽäźĽäžĘåĹĘåÿČåijŔæĺąåđŃïijŇäÿŃéÍćäžčçăĄäÿ■çŽĎis_collective=Trueèąĺçd'ž

```
paddle.enable_static()
configs = X.parse_train_configs()
fleet.init(is_collective=True)
```

åŁäè¡¡æĺąåđŃåŔŁæŢřæ■ő

çŤĺæĹůåŔŕäżěéĂŽè£ĞX.applicationsæÕěåŔčåŁäè¡¡æĹŠäžňéćĎåĚĹåőŽäźĽåě¡çŽĎæĺąåđŃïijŇåę

```
model = X.applications.Resnet50()
downloader = X.utils.Downloader()
local_path = downloader.download_from_bos(
    fs_yaml='https://fleet.bj.bcebos.com/test/loader/small_imagenet.
↪yaml',
    local_path='./data')
batch_size = 32
loader = model.get_train_dataloader(local_path, batch_size=batch_
↪size)
```

åőŽäźĽLocal SGD çŽÿåĚşç■ŰçŢě

çŤĺæĹůéęŰåĚĹéIJĂèęĄåőŽäźĽ'paddle SGD åŕźèśąïijŇåźűåIJĺSGD åŕźèśąäÿ■èő¿ç¡őå■ęäźăçŐĞåŔĆæŢřãĂĆçŽőåĽ■local SGDåŠŇèĞłéĂĆåžŤæ■ěéŢ£ local SGDéČ¡äżĚæŤŕæŇĄSGDåŠŇMomentumäÿd'çğ■äijŸåŇŰåŹĺãĂĆ

- åIJĺ**post Local SGD** äÿ■ïijŇæIJL'äÿd'äÿłçŤĺæĹůèő¿ç¡őåŔĆæŢř `begin_step` åŠŇ `k_steps`ïijŇåśĂéČĺæŽt'æŰřåŠŇåŔĆæŢřåŘŇæ■ěéČ¡çŤśæąĘæđűèĞłåŁĺåőŇæĹŘãĂĆbegin_step æŇĞåőŽäżŐçňňåĞääÿłstepäźŇåŘŐè£ŻèąŇlocal SGDçőŮæşŢïijŇåŔŰåĂijäÿžåd'ğäžŐ0çŽĎæŢt'æŢřïi æŇĞåőŽèő■çżČè£ĞçĺŃäÿ■çŽĎåĚĺåśĂåŔĆæŢřæŽt'æŰřéŮt'éŽŤïijŇåŔŰåĂijäÿžåd'ğäžŐ0çŽĎæŢt'æŢ

```
dist_strategy = fleet.DistributedStrategy()
dist_strategy.localsgd = True
dist_strategy.localsgd_configs = {
    "k_steps": 1,
    "begin_step": 1,
}

optimizer = fluid.optimizer.SGD(learning_rate=0.01)
optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)
```

- åIJĺ **èĞłéĂĆåžŤæ■ěéŢ£ local SGD** äÿ■ïijŇæIJL'äÿd'äÿłçŤĺæĹůèő¿ç¡őåŔĆæŢř `begin_step` åŠŇ `init_k_steps`ãĂĆbegin_step æŇĞåőŽäżŐçňňåĞääÿłstepäźŇåŘŐè£ŻèąŇèĞłéĂĆåžŤlocal SGDçőŮæşŢïijŇåŘŰåĂijäÿžåd'ğäžŐ0çŽĎæŢt'æŢřïijŻçŤĺæĹůéIJĂèęĄèő¿ç¡őinit_k_stepsä¡IJäÿžçňňäÿ éIJĂèęĄæşĺæĎŔçŽĎæŸŕèĞłéĂĆåžŤæ■ěéŢ£ç■ŰçŢěäÿ■ïijŇçşżçż§äijŽźŸèőd'éŹŘåĹűæIJĂåd'ğçŽĎå 16 stepïijŇå¡ŞåĚňåijŔèőąçőŮåĞžçŽĎéŮt'éŽŤåd'ğäžŐ16 æŮűïijŇæŇL'16 steps è£ŻèąŇåŔĆæŢřåŘŇæ■ěãĂĆ

```
dist_strategy = fleet.DistributedStrategy()
dist_strategy.adaptive_localsgd = True
dist_strategy.adaptive_localsgd_configs = {
    "init_k_steps": 1,
```

(äÿŃéąţçżğçż■)

(çż■äÿŁéąţ)

```
    "begin_step": 1,
}

optimizer = fluid.optimizer.SGD(learning_rate=0.01)
optimizer = fleet.distributed_optimizer(optimizer, dist_strategy)
optimizer.minimize(model.loss)
```

åijĂåğŃèő■çżČ

è£ŹäÿĂéČÍåĹĘåŠŇFleet äÿ■åĔűäžŰäżżåŁąå§žæIJňçŻÿåŘŇ:

```
place = fluid.CUDAPlace(int(os.environ.get('FLAGS_selected_gpus',␣
↪0)))
exe = fluid.Executor(place)
exe.run(fluid.default_startup_program())

for i, data in enumerate(loader()):
    start_time = time.time()
    cost_val = exe.run(model.main_prog,
                       feed=data,
                       fetch_list=[model.loss.name])

    end_time = time.time()
    print(
        "worker_index: %d, step%d cost = %f, speed: %f"
        % (fleet.worker_index(), i, cost_val[0], batch_size / (end_
↪time - start_time)))
```

è£ŘèąŇèő■çżČèĎŽælJň

äÿĂèąŇåŘŕåŁÍå■ŢæIJžåď'Žå■ąåĹĘåÿČåijŔèő■çżČïijŽ

```
fleetrun --gpus 0,1,2,3,4,5,6,7 --log_dir log resnet50_localsgd.py

# reader shuffle seed 0
# trainerid, trainer_count 0 8
# read images from 0, length: 160146, lines length: 160146, total:␣
↪1281168
# worker_index: 0, step0 cost = 7.151402, speed: 37.698432
# worker_index: 0, step1 cost = 7.112389, speed: 101.518513
# worker_index: 0, step2 cost = 7.004275, speed: 111.062341
# worker_index: 0, step3 cost = 7.039385, speed: 62.173126
# worker_index: 0, step4 cost = 6.985911, speed: 104.058060
# ......
```

CHAPTER 20

éčđæąĺåĹĘåÿČåijŔèő■çżČå§žçž£æŁěåŚŁ

CHAPTER 21

Resnet50æĂğèČ¡å§žåĞĘ

Resnet50æŸŕå¡ŞåĽ■èğĘèğĽéćĘå§§æŕŤè¿ČéĂŽçŤĺçŽĎéćĎèő■çżČæĺąåđŃåŘŐçńŕïijŇåŘŇæŮűäź§ä¡IJä

21.1 è¡ŕçąňäžűéĚ■ç¡őæČĚåĘţ

21.1.1 å§žælJňçĽĹælJňä£ąæĄŕ

è¡ŕçąňäžűæŇĞæäĞ	åĚůä¡ŞéĚ■ç¡ő
åőđä¿Ńçśžåđŕ	çŹ¿åžęX-Man 2.0
å■Ţåőđä¿ŃGPU	8x NVIDIAÂő TeslaÂő V100
æŞ■ä¡IJçşżçż§	Ubuntu 16.04 LTS with tests run via Docker
CPU	Intel(R) Xeon(R) Gold 6271C CPU @ 2.60GHz
åĘĚå■Ÿ	512G
CUDA / CUDNNçĽĹælJň	10.1 / 7.6.5
NCCL / DALI çĽĹælJň	2.4.7 / 0.24.0
åď ŽGPUåőđä¿ŃäžŠèĄŤä£ąæĄŕ	InfiniBand 100 Gb/sec
Paddle Github Commit	
FleetX Github Commit	
çąňçŻŸçśžåđŕ	æIJňåIJřSSDçąňçŻŸ
æŢřæ■őéŻĘ	ImageNet
èŕĎäijřæĺąåđŕ	Resnet50
åď■çŐřäżčçăĄåIJřåİĂ	Resn et50-Benchmark
PythonçĽĹælJň	3.7

21.1.2 çąňäżűæŃŞæĽŚ

```
nvidia-smi topo -m
```

```
GPU0    GPU1    GPU2    GPU3    GPU4    GPU5    GPU6    GPU7    ␣
↪mlx5_0  CPU Affinity
GPU0     X      NV2     NV2     NV1     NV1     NODE    NODE    ␣
↪NODE    NODE    0-23
GPU1    NV2      X      NV1     NV1     NODE    NV2     NODE    ␣
↪NODE    NODE    0-23
GPU2    NV2     NV1      X      NV2     NODE    NODE    NV1     ␣
↪NODE    NODE    0-23
GPU3    NV1     NV1     NV2      X      NODE    NODE    NODE    NV2␣
↪    NODE    0-23
GPU4    NV1     NODE    NODE    NODE     X      NV2     NV2     NV1␣
↪    NODE    0-23
GPU5    NODE    NV2     NODE    NODE    NV2      X      NV1     NV1␣
↪    NODE    0-23
GPU6    NODE    NODE    NV1     NODE    NV2     NV1      X      NV2␣
↪    NODE    0-23
GPU7    NODE    NODE    NODE    NV2     NV1     NV1     NV2      X ␣
↪    NODE    0-23
mlx5_0  NODE    NODE    NODE    NODE    NODE    NODE    NODE    ␣
↪NODE     X

Legend:

X    = Self
SYS  = Connection traversing PCIe as well as the SMP interconnect␣
↪between NUMA nodes (e.g., QPI/UPI)
NODE = Connection traversing PCIe as well as the interconnect␣
↪between PCIe Host Bridges within a NUMA node
PHB  = Connection traversing PCIe as well as a PCIe Host Bridge␣
↪(typically the CPU)
PXB  = Connection traversing multiple PCIe switches (without␣
↪traversing the PCIe Host Bridge)
PIX  = Connection traversing a single PCIe switch
NV#  = Connection traversing a bonded set of # NVLinks
```

21.2 æĂğèČ¡æţŃèŕŢæŰźæşŢ

- çąňäżűèţĎæžŘ éĞĞçŤĺåď'ŽæIJžåď'Žå■ąèő■çżČïijŇäžěåőđä¿ŃæŢř x å■Ţåőđä¿ŃGPUå■ąæŢřä¡IJäÿžèŕĎäžůæăĞåĞĘïijŇèŕĎäžů `1 x 1`, `1 x 8`, `2 x 8`, `4 x 8`æČĚåĘţäÿŃçŽĎæĂğèČ¡å§žåĞĘãĂĆ
- èő■çżČèűĚåŔĆæŢř æL'źéĞŔåď'ğåřŔïijĹBatch SizeïijL'åŕźèő■çżČæĂğèČ¡å¡śåŞ■æIJĂåď'ğïijŇåŻăæ■d'
- æţŃèŕŢæŇĞæăĞèŐůåŔŰæŰźæşŢ å¡ŞåL'■äÿżæţAçŽĎæůśåžęå■ęäźăæąĘæđűéĂŽåÿÿéĞĞçŤĺåijĆæ■ěæ

21.3 å§žåĞĘæţŃèŕŢçżŞædIJ

- å■Ţä¡■ïijŽImages/sïijŇä¡£çŤÍçš¿åžęFP32ïijŇDistributedStrategyåęĆäÿŃïijŽ

```
import paddle
import paddle.distributed.fleet as fleet
dist_strategy = fleet.DistributedStrategy()
exec_strategy = fluid.ExecutionStrategy()
exec_strategy.num_threads = 2
exec_strategy.num_iteration_per_drop_scope = 100
dist_strategy.execution_strategy = exec_strategy
build_strategy = fluid.BuildStrategy()
build_strategy.enable_inplace = False
build_strategy.fuse_elewise_add_act_ops = True
build_strategy.fuse_bn_act_ops = True
dist_strategy.build_strategy = build_strategy
dist_strategy.nccl_comm_num = 1
```

batch / node	1 x 1	1 x 8	2 x 8	4 x 8
32	335.43	2488.49	4629.71	9093.41
64	353.38	2643.75	5325.44	10536.83
128	368.11	2797.31	5635.98	11261.72

- å■Ţä¡■ïijŽImages/sïijŇä¡£çŤÍèĞłåŁÍæůůåŘĹçš¿åžęAutomatic Mixed Precision(AMP)è£ŻèąŇèő■çżĆïijŇDistributedStrategyåęĆäÿŃïijŽ

```
import paddle
import paddle.distributed.fleet as fleet
dist_strategy = fleet.DistributedStrategy()
exec_strategy = fluid.ExecutionStrategy()
exec_strategy.num_threads = 2
exec_strategy.num_iteration_per_drop_scope = 100
dist_strategy.execution_strategy = exec_strategy
build_strategy = fluid.BuildStrategy()
build_strategy.enable_inplace = False
build_strategy.fuse_elewise_add_act_ops = True
build_strategy.fuse_bn_act_ops = True
dist_strategy.build_strategy = build_strategy
dist_strategy.amp = True
dist_strategy.nccl_comm_num = 1
```

batch / node	1 x 1	1 x 8	2 x 8	4 x 8
32	740.01	4467.82	8628.19	16970.01
64	919.95	6148.98	12071.29	23682.78
128	1018.3	7324.31	14342.03	28397.43
256	1096.5	8166.11	16189.79	32366.39

- å■Ţä¡■ïijŽImages/s,èĞłåŁĺåźűèąŇæĺąåijŕïijŇDistributedStrategyåęĆäÿŃïijŽ

```
import paddle.distributed.fleet as fleet
dist_strategy = fleet.DistributedStrategy()
dist_strategy.auto = True
```

äÿžäžĘèŐůå¿ÛæŻťåě¡çŽĎæĂğèČ¡ïijŇæĹŚäżňéżŸèőď'æĽŞåijĂäžĘDALIè£ŻèąŇæŢřæ■őIOïijŇè£ŹéĞ

batch / node	1 x 1	1 x 8	2 x 8	4 x 8
32	666.38	4467.82	8711.69	19107.42
64	761	6148.98	12076.77	24314.58
128	890.03	6793.73	13514.66	27277.36
256	938.57	7305.66	14599.55	29361.24

CHAPTER 22

BertæÍąåđŃèő■çżČæĂğèČ¡å§žçž£

CHAPTER 23

TransformeræĺąåđŃæĂğèČ¡å§žçž£

CHAPTER 24

VGG16æĺąåđŃèő■çżČæĂğèČ¡å§žçž£

VGG16æŸŕå¡ŞåL’■èğĘèğL’écĘå§§æŕŤè¿ČéĂŽçŤĺçŽĎécĎèő■çżČæĺąåđŃåŘŐçńŕïijŇåŘŇæŮűäź§ä¡IJäÿ

24.1 è¡ŕçąňäžűéĚ■ç¡őæČĚåĘţ

24.1.1 å§žælJňçL’ĹælJňä£ąæĄŕ

è¡ŕçąňäžűæŇĞæăĞ	åĚůä¡ŞéĚ■ç¡ő
åőđä¿ŃçśžåđŃ	çŹ¿åžęX-Man 2.0
å■Ţåőđä¿ŃGPU	8x NVIDIAÂő TeslaÂő V100
æŞ■ä¡IJçşżçż§	Ubuntu 16.04 LTS with tests run via Docker
CPU	Intel(R) Xeon(R) Gold 6271C CPU @ 2.60GHz
åĘĚå■Ÿ	512G
CUDA / CUDNNçL’ĹælJň	10.1 / 7.6.5
NCCL / DALI çL’ĹælJň	2.4.7 / 0.24.0
åď’ŽGPUåőđä¿ŃäžŠèĄŤä£ąæĄŕ	InfiniBand 100 Gb/sec
Paddle Github Commit	b6711e24ea5e17fa24e9b1ba516fe03186dd4a2c
FleetX Github Commit	296e0d71a33a8d250f6626733bc2535465f5f6e0
çąňçŻŸçśžåđŃ	æIJňåIJřSSDçąňçŻŸ
æŢřæ■őéŽĘ	ImageNet
èŕĎäijřæĺąåđŃ	VGG16
åď’■çŐřäžčçăĄåIJřåİĂ	VGG16 -Benchmark
PythonçL’ĹælJň	3.7

24.1.2 çąňäżűæŃŞæĽŚ

```
nvidia-smi topo -m
```

```
GPU0    GPU1    GPU2    GPU3    GPU4    GPU5    GPU6    GPU7    ␣
↪mlx5_0  CPU Affinity
GPU0     X      NV2     NV2     NV1     NV1     NODE    NODE    ␣
↪NODE    NODE    0-23
GPU1    NV2      X      NV1     NV1     NODE    NV2     NODE    ␣
↪NODE    NODE    0-23
GPU2    NV2     NV1      X      NV2     NODE    NODE    NV1     ␣
↪NODE    NODE    0-23
GPU3    NV1     NV1     NV2      X      NODE    NODE    NODE    NV2␣
↪    NODE    0-23
GPU4    NV1     NODE    NODE    NODE     X      NV2     NV2     NV1␣
↪    NODE    0-23
GPU5    NODE    NV2     NODE    NODE    NV2      X      NV1     NV1␣
↪    NODE    0-23
GPU6    NODE    NODE    NV1     NODE    NV2     NV1      X      NV2␣
↪    NODE    0-23
GPU7    NODE    NODE    NODE    NV2     NV1     NV1     NV2      X ␣
↪    NODE    0-23
mlx5_0  NODE    NODE    NODE    NODE    NODE    NODE    NODE    ␣
↪NODE     X

Legend:

X    = Self
SYS  = Connection traversing PCIe as well as the SMP interconnect␣
↪between NUMA nodes (e.g., QPI/UPI)
NODE = Connection traversing PCIe as well as the interconnect␣
↪between PCIe Host Bridges within a NUMA node
PHB  = Connection traversing PCIe as well as a PCIe Host Bridge␣
↪(typically the CPU)
PXB  = Connection traversing multiple PCIe switches (without␣
↪traversing the PCIe Host Bridge)
PIX  = Connection traversing a single PCIe switch
NV#  = Connection traversing a bonded set of # NVLinks
```

24.2 æĂğèČ¡æţŃèŕŢæŰźæşŢ

- çąňäżűèţĎæžŘ éĞĞçŤĺåď'ŽæIJžåď'Žå■ąèő■çżČïijŇäžěåőđä¿ŃæŢř x å■Ţåőđä¿ŃGPUå■ąæŢřä¡IJäÿžèŕĎäžůæăĞåĞĘïijŇèŕĎäžů `1 x 1`, `1 x 8`, `2 x 8`, `4 x 8`æČĚåĘţäÿŃçŽĎæĂğèČ¡å§žåĞĘãĂĆ
- èő■çżČèűĚåŔĆæŢř æĽźéĞŔåď'ğåřŔïijĹBatch SizeïijĽ'åŕźèő■çżČæĂğèČ¡å¡śåŞ■æIJĂåď'ğïijŇåŽăæ■d'ïijŇå■şäÿ■èĂČèŹŠæŤűæŢŽçŽĎåŕźæŕŤãĂĆ

- æţŃèŕŢæŇĞæăĞèŐůåŔŰæŰźæşŢ å¡ŞåĽ■äÿžæţAçŽĎæůšåžęå■ęäźăæąĘæđűéĂŽåÿÿéĞĞçŤĺåijĆæ■ěæéÍć100äÿłstepçŽĎåźşåİĞåŘđåŘŘä¡IJäÿžå■Ţæňąäżżå£qçŽĎèő■çżĈåŘđåŘŘãĂĆäÿžäžĘéĄ£åĚ■çąňäžűć

24.3 å§žåĞĘæţŃèŕŢçżŞæđIJ

- å■Ţä¡■ïijŽImages/sïijŇä¡£çŤĺçš¿åžęFP32ïijŇDistributedStrategyåęĆäÿŃïijŽ

```
exec_strategy = fluid.ExecutionStrategy()
dist_strategy = fleet.DistributedStrategy()
exec_strategy.num_threads = 2
exec_strategy.num_iteration_per_drop_scope = 100
dist_strategy.execution_strategy = exec_strategy
build_strategy = fluid.BuildStrategy()
build_strategy.enable_inplace = False
build_strategy.fuse_elewise_add_act_ops = True
build_strategy.fuse_bn_act_ops = True
dist_strategy.build_strategy = build_strategy
dist_strategy.nccl_comm_num = 1
```

batch / node	1 x 1	1 x 8	2 x 8	4 x 8
32	252.68	1888.20	2873.18	5548.48
64	261.33	1980.04	3900.12	7617.47
128	266.24	2027.06	4028.78	7848.70

- å■Ţä¡■ïijŽImages/sïijŇä¡£çŤĺèĞłåŁĺæůůåŘĹçš¿åžęAutomatic Mixed Precision(AMP)è£ŻèąŇèő■çżČïijŇDistributedStrategyåęĆäÿŃïijŽ

```
import paddle
import paddle.distributed.fleet as fleet
dist_strategy = fleet.DistributedStrategy()
exec_strategy = fluid.ExecutionStrategy()
exec_strategy.num_threads = 2
exec_strategy.num_iteration_per_drop_scope = 100
dist_strategy.execution_strategy = exec_strategy
build_strategy = fluid.BuildStrategy()
build_strategy.enable_inplace = False
build_strategy.fuse_elewise_add_act_ops = True
build_strategy.fuse_bn_act_ops = True
dist_strategy.build_strategy = build_strategy
dist_strategy.amp = True
dist_strategy.nccl_comm_num = 1
```

batch / node	1 x 1	1 x 8	2 x 8	4 x 8
32	407.69	3332.17	5136.50	9544.81
64	468.51	3708.32	7112.45	14013.01
128	512.02	3892.58	7618.34	15219.57
256	439.47	3409.96	6779.20	13443.23

- å■Ţä¡■ïijŽImages/s, èĞłåŁĺåźűèąŇæĺąåijŕïijŇDistributedStrategyåęĆäÿŃïijŽ

```
import paddle.distributed.fleet as fleet
dist_strategy = fleet.DistributedStrategy()
dist_strategy.auto = True
```

äÿžäžĘèŐůå¿ŮæŻť'åě¡çŽĎæĂğèČ¡ïijŇæĹŚäżňéżŸèőd'æĽŞåijĂäžĘDALIè£ŻèąŇæŢřæ■őIOïijŇè£ŹéĞ

batch / node	1 x 1	1 x 8	2 x 8	4 x 8
32	409.68	3044.60	4840.74	7668.70
64	455.98	3395.67	6525.20	12237.04
128	472.81	3587.29	7019.13	13562.80
256	407.88	3154.15	6217.92	12147.46

CHAPTER 25

Word2vecæĺąåđŃæĂğèČ¡å§žåĞĘ

word2vecèćńåź£æşŻè£ŘçŤĺäžŐNLPåŔŁæŐĺè■Řçşżçż§écĘå§§ïijŇåŘŇæŮűäź§ä¡IJäÿžèŕĎäżůæůśåžęå■
billionæŢřæ■őéŻĘäÿŁçŽĎåĹĘåÿČåijŔèő■çżČçŽĎæĂğèČ¡å§žåĞĘä¿ŻçŤĺæĹůåŔĆèĂČ,åŇĚåŘń4ãĂĄ8ãĂĄ

25.1 è¡ŕçąňäżűéĚ■ç¡őæČĚåĘţ

25.1.1 å§žælJňçĽĹælJňä£ąæĄŕ

è¡ŕçąňäżűæŇĞæăĞ	åĚůä¡ŞéĚ■ç¡ő
åőđä¿Ńçśžåđŕ	çžŕCPUèő■çżČéŻĘç¿d’
æŞ■ä¡IJçşżçż§	Ubuntu 16.04 LTS with tests run via Docker
CPU	Intel(R) Xeon(R) CPU E5-2450 v2 @ 2.50GHz
åĘĚå■Ÿ	128G
Paddle Github Commit	
FleetX Github Commit	
çąňçŻŸçśžåđŕ	æIJňåIJřSSDçąňçŻŸ
æŢřæ■őéŻĘ	1-billion
èŕĎäijřæĺąåđŕ	Work2vec
åd’■çŐřäżčçăĄåIJřåİĂ	Word 2vec-Benchmark
PythonçĽĹæIJň	3.7

25.2 æĂğèČ¡æţŃèŕŢæŰźæşŢ

- çąňäżűèţĎæžŘ éĞĞçŤĺåď ŽæIJžåď Žè£ŽçÍŃèő■çżČïijŇæŕŔäÿĂåŔřæIJ■åŁąåŹĺåőđä¿ŃåİĞåŘŕåŁĺäÿĂä 4, 8, 16, 32æČĚåĘţäÿŃçŽĎæĂğèČ¡å§žåĞĘãĂĆ
- èő■çżČèűĚåŔĆæŢř æL'źéĞŔåď ğářŔïijĹBatch SizeïijL'åŕźèő■çżČæĂğèČ¡å¡śåŞ■æIJĂåď ğïijŇåŽăæ■d'ă
- æţŃèŕŢæŇĞæăĞèŐůåŔŰæŰźæşŢ å¡ŞåL'■äÿżæţAçŽĎæůśåžęå■ęäźăæąĘæđűéĂŽåÿÿéĞĞçŤĺåijĆæ■ěæ

25.3 å§žåĞĘæţŃèŕŢçżŞæđlJ

- å■Ţä¡■ïijŽImages/sïijŇä¡£çŤĺçš¿åžęFP32ïijŇDistributedStrategyåęĆäÿŃïijŽ

```
import paddle
import paddle.distributed.fleet as fleet

dist_strategy = fleet.DistributedStrategy()
dist_strategy.a_sync=True
dist_strategy.a_sync_configs = {"k_steps": 100}
```

batch / node	4	8	16	32
100	55597.5	55082.37	53302.63	47280.91

CHAPTER 26

çŤÍæĹůFAQ

- TBA

FleetXä¡£çŤÍApache License 2.0åijĂæžŘå■Ŕèőő