TensorFlow Distribution

|  |  |
| --- | --- |
| 制作小组 | BIG DATA GROUP |
| 制作日期 | 2018.10.30 |

Contents

[TensorFlow Distribution 3](#_Toc528160751)

[1.Distrbuted Training 3](#_Toc528160752)

[1.1.Property of TensorFlow Distribution 3](#_Toc528160753)

[1.2.Single Process Distribution: 3](#_Toc528160754)

[1.3.Multiple Server Distribution: 3](#_Toc528160755)

# TensorFlow Distribution

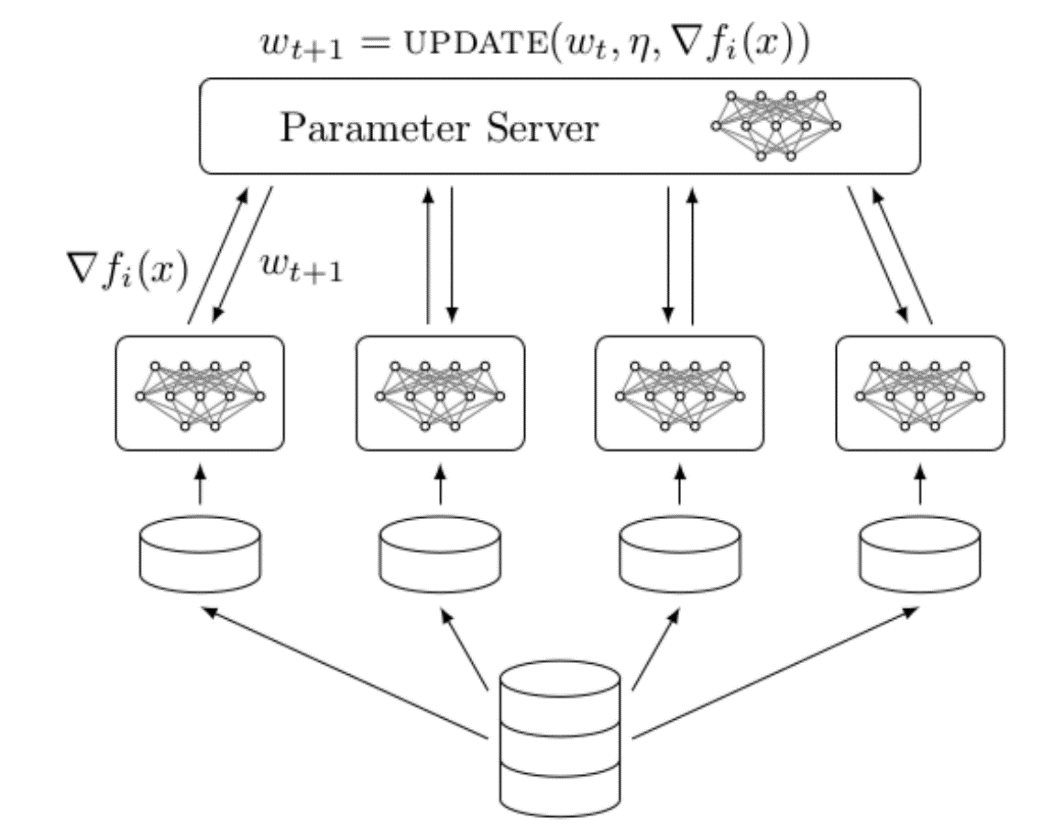
# 1.Distrbuted Training

## 1.1.Property of TensorFlow Distribution

A TensorFlow "cluster" is a set of "**tasks**" that participate in the distributed execution of a TensorFlow graph. Each task is associated with a TensorFlow "**server**", which contains a "**master**" that can be used to **create sessions**, and a "**worker**" that **executes operations** in the graph. A cluster can also be divided into one or more "jobs", where each job contains one or more tasks.

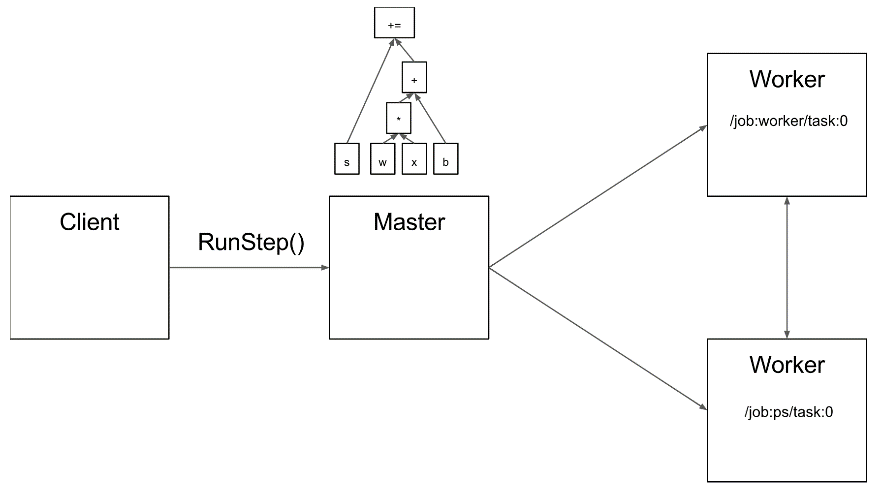
TensorFlow 分布式的特点就是需要指定手动对应server和worker的host，并且可以细粒度的指定变量或是运算分别需要在哪台机器运行，灵活度更高。

TensorFlow分布式模型训练



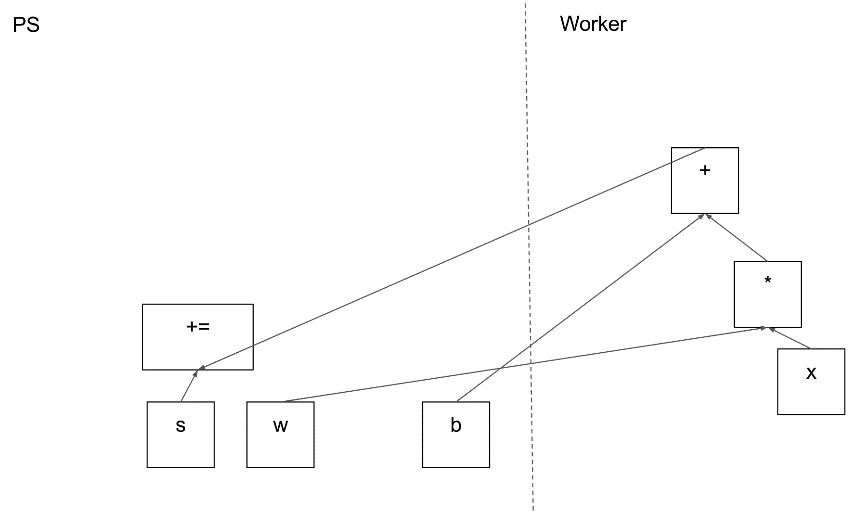
分布式架构

分布式计算w\*x+b, client发送代码至master

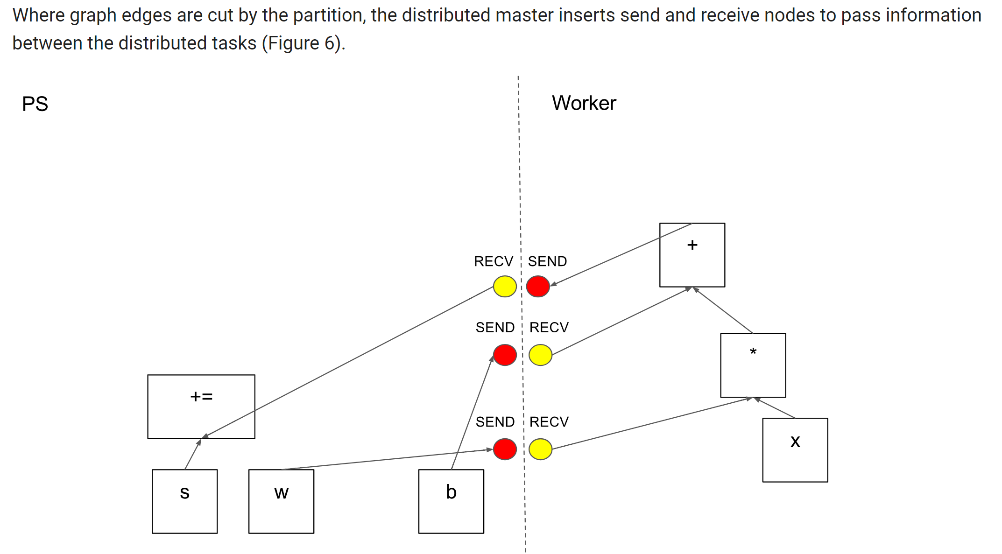


分布式通讯

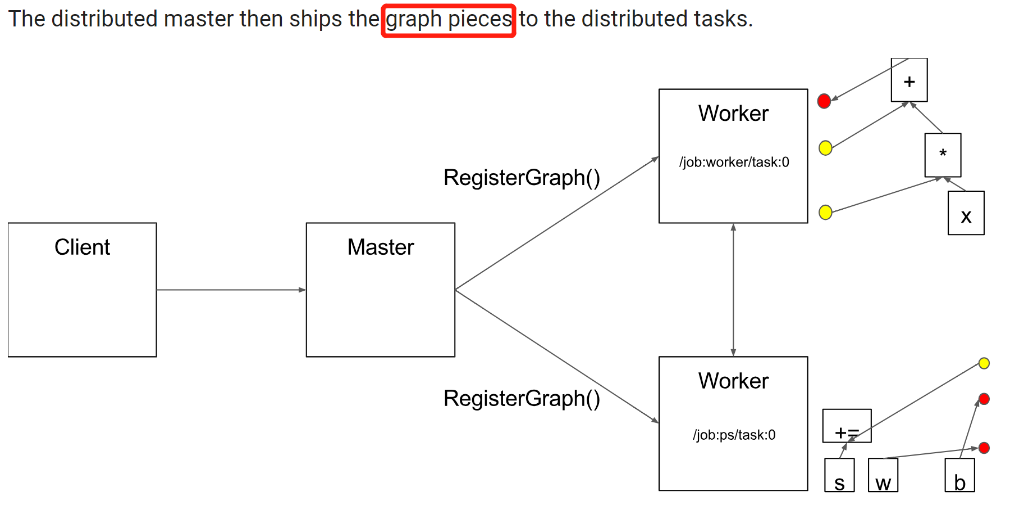
PS:变量存储及最终结果运算



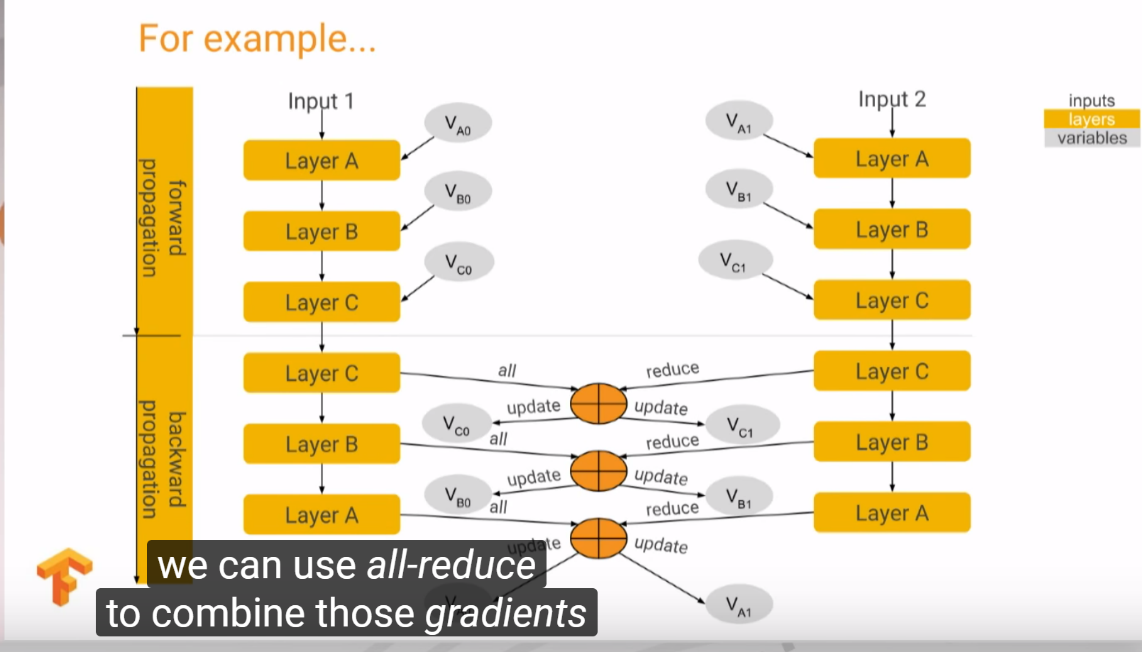
PS与worker进行通信



Master与两个worker分别进行通信及worker进行运算



Deep model for distributed model training:



## 1.2.Single Process Distribution:

**Code example:**

import tensorflow as tf

c = tf.constant("Hello, distributed TensorFlow!")

server = tf.train.Server.create\_local\_server()

sess = tf.Session(server.target) # Create a session on the server.

sess.run(c)

## 1.3.Multiple Server Distribution:

Code example:

import tensorflow as tf

import sys

host1 = 'localhost:1002'

host2 = 'localhost:1003'

# Get which task to process from terminal

task\_num = int(sys.argv[1])

# Make a cluster with localhost for 2 web addresses

cluster = tf.train.ClusterSpec({'local': [host1, host2]})

# Here is used to build a server, task\_index is used for which task to be placed on.

server = tf.train.Server(cluster, job\_name='local', task\_index=task\_num)

print('Starting server :{}'.format(task\_num))

sess = tf.Session(server.target)

a = tf.constant(2)

### This is not to specify which operation to run on which place.

b = a - 10

c = a + 12

out = b + c

print('\*'\*20, sess.run(out))

# Start the server and join the service

server.start()

server.join()

在全部代码写入在dis\_exa.py文件中之后，分别打开两个终端并进入dis\_exa.py目录中：

在终端1中输入：python dis\_exa.py 0

在终端2中输入：python dis\_exa.py 1

则可以分别在终端中看到对应的输出结果。

Noted:对于变量或是运算，可以分别进行指定执行server, code example 如下(只需更改上面代码中的变量运算代码)：

### Here I can also specify which variable running place

with tf.device('/job:local/task:0'):

b = a - 10

with tf.device('/job:local/task:1'):

c = a + 12

out = b + c

with tf.Session('grpc://'+ host1) as sess: ## Here can also use some service to run the session

print('\*'\*20)

print('Final result:', sess.run(out)) ## Result is also with be printed on task1 for host2

过程同上，可以在终端中看到对应的结果(最终的运行结果只会在host2终端上看到！)

Noted:如果出现了报错Could not start gRPC server，则为端口专用，重新更换端口即可。