



caicloud
才云

Caicloud TaaS Introduction

TensorFlow on Kubernetes

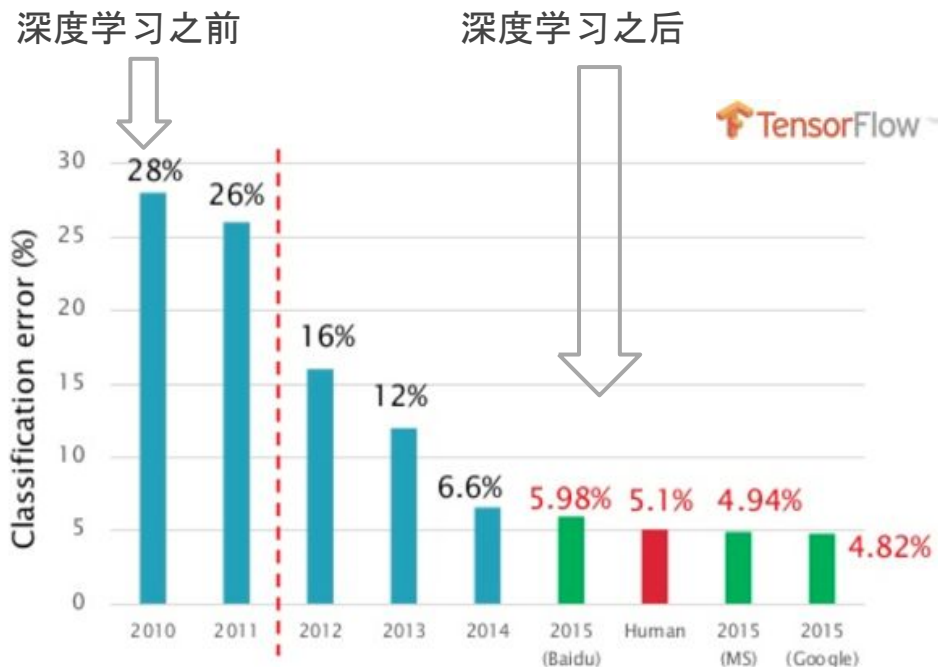
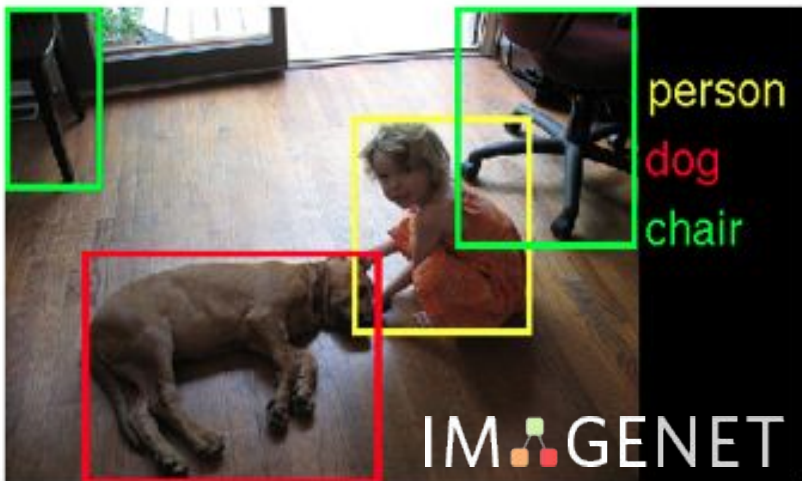
岑鹏浩 才云科技

Topics

- Introduction to Deep Learning
- Introduction to TensorFlow
- TensorFlow on Kubernetes
- Introduction to TaaS
- TaaS Demo









- Open Source by Google
- Python and C++ API
- Distributed Processing
- Supports CPUs & GPUs
- Machine Learning & Deep Neural Network
- Based on data flow graphs



- `docker run -p 8888:8888 -p 6006:6006 cargo.caicloud.io/tensorflow/tensorflow:1.0.0`

Jupyter 编辑器端口

TensorBoard 端口

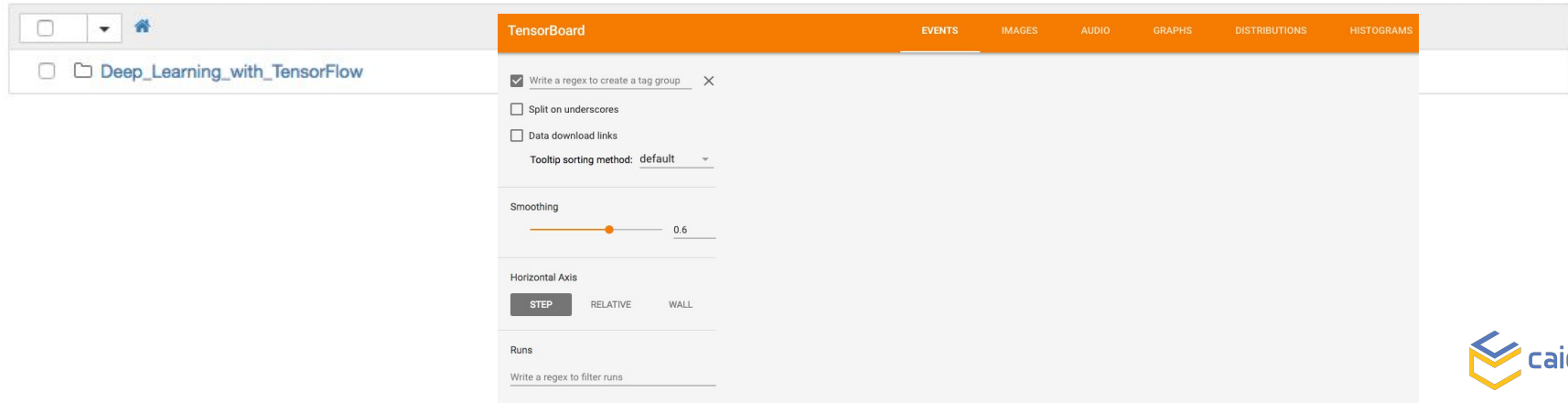


Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↕ ↻



The screenshot shows the Jupyter TensorBoard interface. On the left, there is a file browser showing a folder named "Deep_Learning_with_TensorFlow". The main area is titled "TensorBoard" and contains several configuration options:

- Write a regex to create a tag group
- Split on underscores
- Data download links
- Tooltip sorting method: default
- Smoothing: 0.6
- Horizontal Axis: STEP (selected), RELATIVE, WALL
- Runs: Write a regex to filter runs

```
1 #coding=utf-8
2 #
3 #Copyright 2017 caicloud authors. All rights reserved.
4 #
5
6 import tensorflow as tf
7
8 # tensorflow version 1.1.0
9 print tf.__version__
10
11 # tensorflow 通过 session 维护上下文, 所有执行都需要通过 session
12 session = tf.InteractiveSession()
13
14 with tf.name_scope('input'):
15     ... # 数据都存储在 "tensor" 中
16     ... input1 = tf.constant([1.0, 2.0, 3.0], name="input1")
17     ... # 变量都存储在 "variable" 中
18     ... input2 = tf.Variable(tf.random_uniform([3]), name="input2")
19
20 # 在运行前, 所有变量都需要初始化
21 tf.global_variables_initializer().run()
22
23 with tf.name_scope('add'):
24     ... output = tf.add(input1, input2, name="add")
25
26 # 把日志文件写入本地
27 writer = tf.summary.FileWriter("./log", session.graph)
28
29 # 所有结果都需要先运行才能获取
30 print output.eval()
```

session

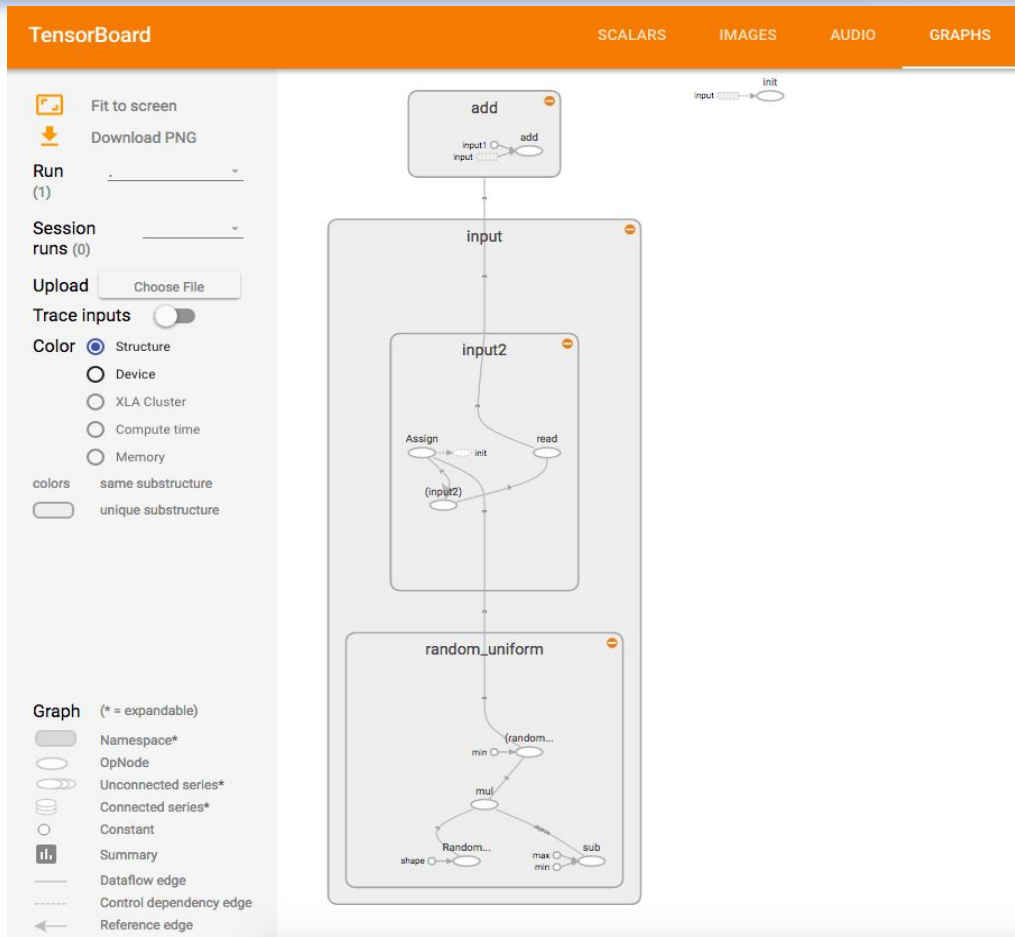
name_scope

constant

variable

\$ tensorboard --logdir ./log/

Starting TensorBoard 47 at http://0.0.0.0:6006

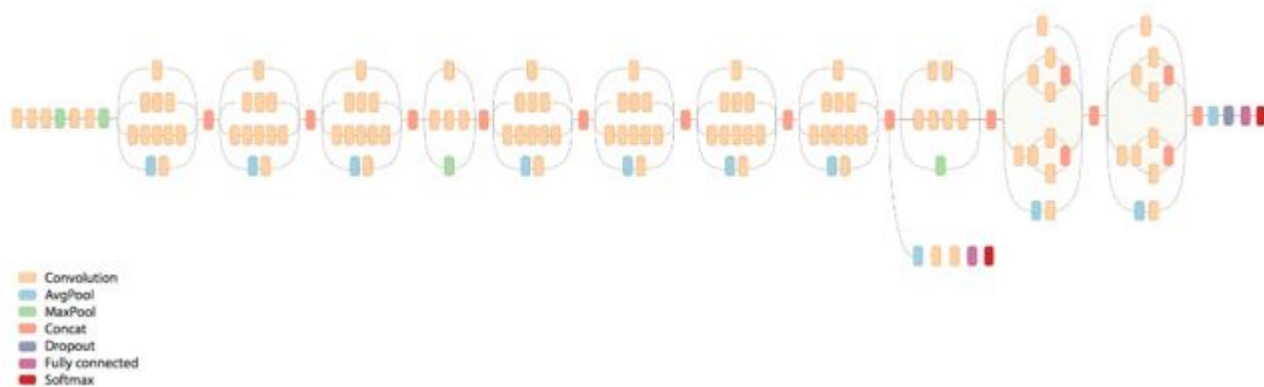


The screenshot displays the TensorBoard interface with an orange header bar. On the left, there is a sidebar with various controls: 'Fit to screen', 'Download PNG', 'Run (1)', 'Session runs (0)', 'Upload Choose File', 'Trace inputs' (toggle), 'Color' (Structure selected), and 'colors' (same substructure and unique substructure options). Below these is a 'Graph' legend with symbols for Namespace*, OpNode, Unconnected series*, Connected series*, Constant, Summary, Dataflow edge, Control dependency edge, and Reference edge. The main area shows a computational graph with three main namespaces: 'add' at the top, 'input' in the middle, and 'random_uniform' at the bottom. The 'add' node has two inputs, 'input1' and 'input'. The 'input' namespace contains an 'input2' namespace with 'Assign' and 'read' nodes, and an '(input2)' node. The 'random_uniform' namespace contains a 'mul' node, a 'Random...' node, a 'min' node, a 'max' node, and a 'sub' node. The graph is connected by dataflow edges.

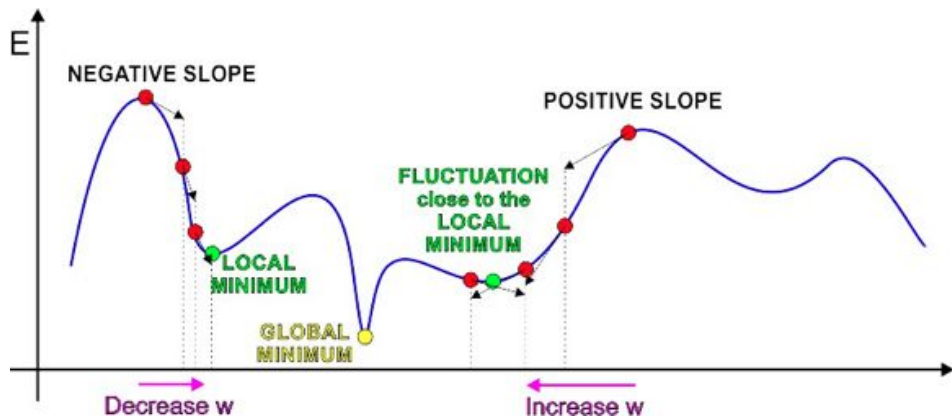
- 没有免费的午餐 -- 计算量问题

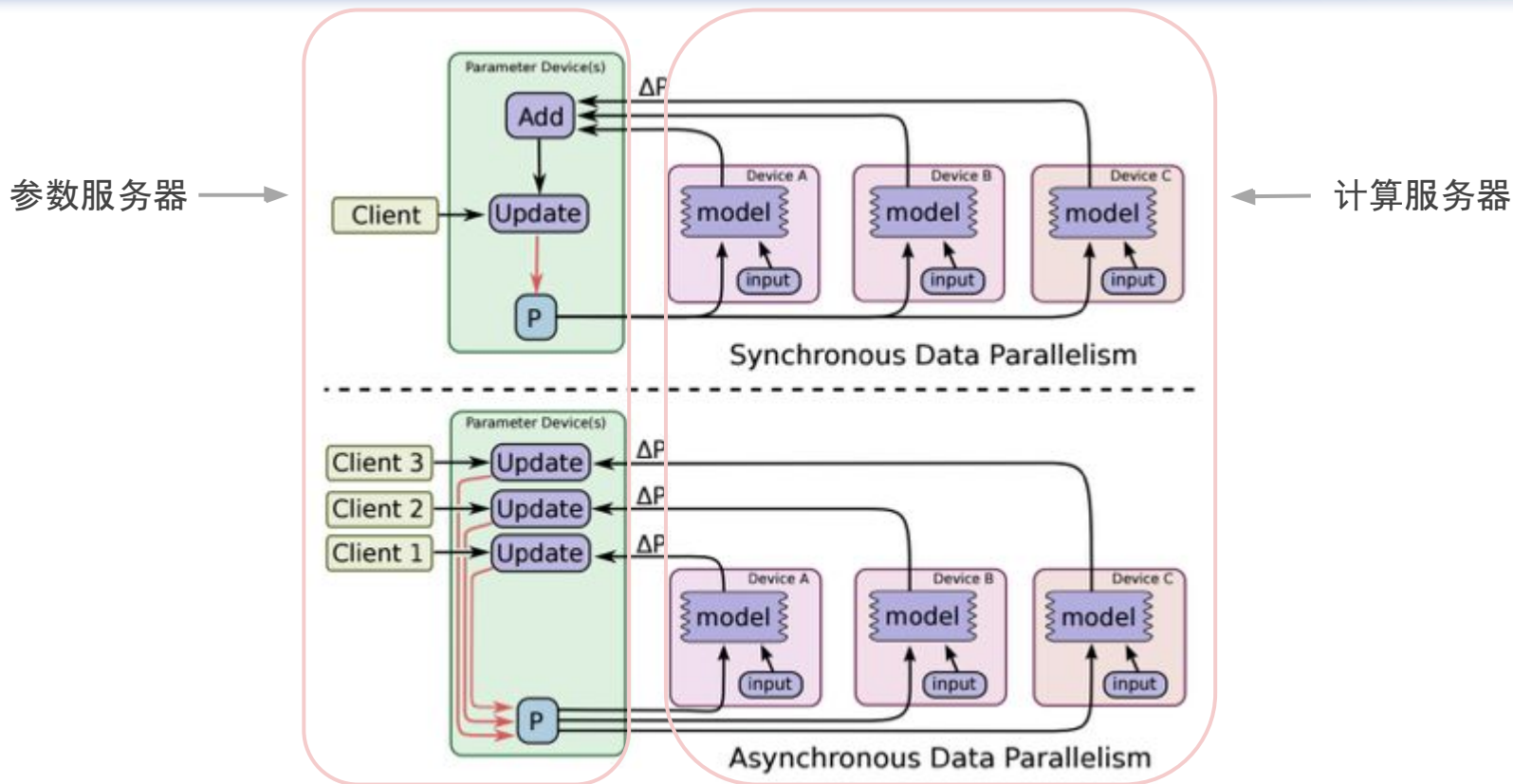
Inception-v3 model for ImageNet

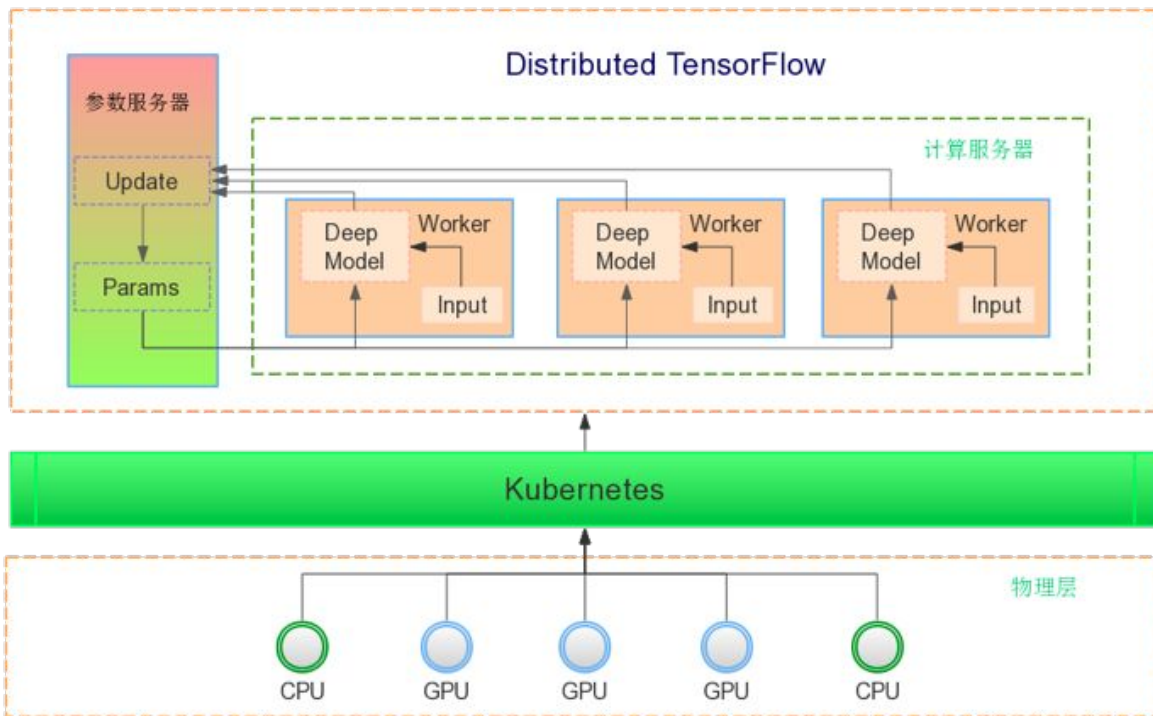
- 2500 万参数
- 每次 inference/forward prorate 50亿 乘法/加法 操作



- 没有免费的午餐 -- 优化问题
 - 神经网络结构复杂, 难以直接求解
 - 迭代优化算法 -- 梯度下降法
 - 需要海量数据以及海量计算
 - 单机需要训练达到 78% 的正确率需要半年







集群管理

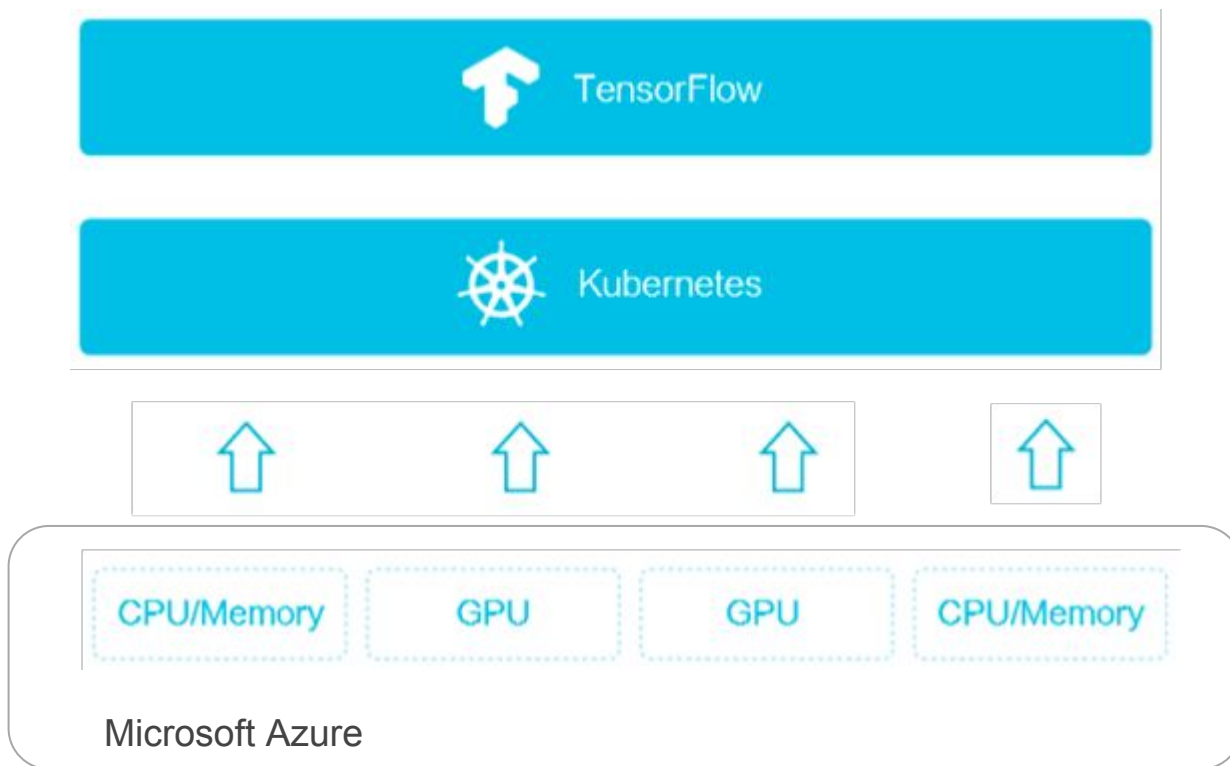
- 通过 Kubernetes 的 Service 做服务发现
- 通过 Replica Controller 控制失败重启
- Kubernetes 提供监控、调度等功能

生命周期管理

- 目前不会自动结束
- 难以区分正常结束还是异常退出
- 手动管理

存储解决方案

- 使用 GlusterFS、ceph 等分布式存储
- 通过 Kubernetes 的 PV/PVC 做持久化存储



操作	原生态 TensorFlow	Caicloud TaaS
安装方式	通过 pip 或者源代码安装 或者通过Docker启动	无需安装, 直接使用 TensorFlow 环境
分布式环境设置	需要登录每一台机器安装环境	自动环境配置
GPU 管理	人工协调, 没有管理机制	提供自动 GPU 管理机制, 并且支持多租户
资源分配	人工协调, 没有管理机制	通过 Kubernetes 自动管理, 并且支持多租户

操作	原生态 TensorFlow	Caicloud TaaS
编写 TensorFlow 程序	手动设置分布式集群 手动管理集群资源	提供 TensorFlow 高层封装 省去分布式集群配置、管理工作
启动 TensorFlow 训练任务	手动登录到每台机器 启动训练任务	通过图形化界面配置训练任务 一键启动分布式训练任务
资源监控	手动登录到每台机器 人工监控	提供统一的资源监控界面 集群资源一目了然
任务管理	手动管理训练过程中的 日志、模型文件	基于项目的自动管理 方便的 UI 界面

<https://taas.caicloud.io>

Q&A



caicloud
才云

谢谢大家！