深度学习平台调研工作

陈志凌 2018-07-09

1.深度学习平台调研工作

2.Tensorflow实践

3.对平台的设想

4.接下来的计划

国内外的深度学习平台

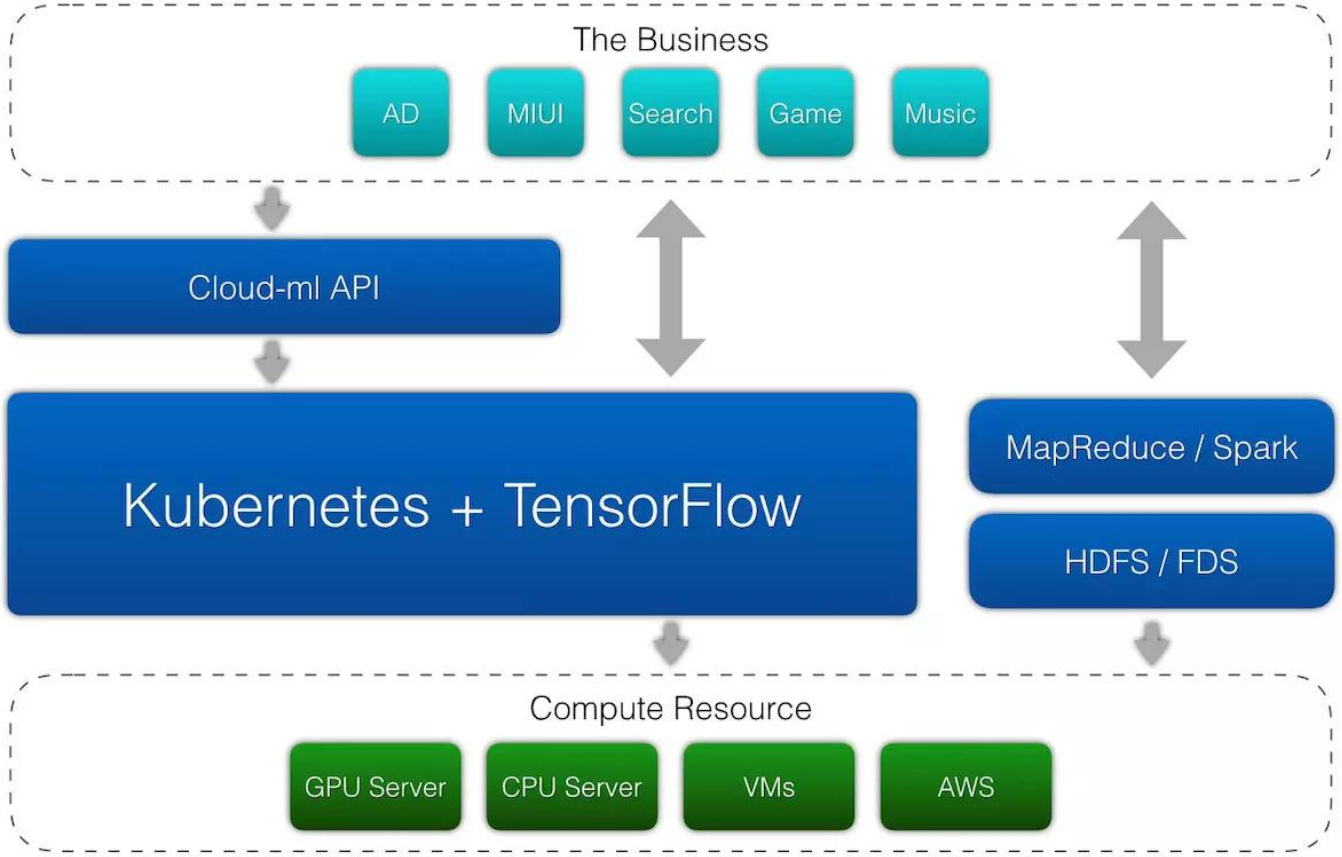
调研工作

- 主要通过增林哥提供的资料/Google/Baidu/知乎/掘金/大咖说等
- 主要收集对应平台的架构内容和技术实现
- 本次主要介绍几个：
  - 小米cloud-ml
  - Azure DSVM
  - 七牛云AVA
  - 饿了么elearn

小米cloud-ml

- 简介
  - 小米云深度学习服务，简称Xiaomi Cloud-ML，是小米生态云针对机器学习优化的高性能、分布式云服务
  - 功能：
    - 在云端使用GPU集群训练模型，秒级启动分布式训练任务
    - 兼容TensorFlow等深度学习框架
    - 提供模型开发、训练、调优、测试、部署和预测一站式解决方案

cloud-ml架构


The Business
AD
MIUI
Search
Game
Music
Cloud-ml API
Kubernetes + TensorFlow
MapReduce / Spark
HDFS / FDS
Compute Resource
GPU Server
CPU Server
VMs
AWS

Xiaomi Cloud-ML特性

易用性

支持CLI，可在Linux/Mac/Windows操作系统或者Docker中运行，提供API、SDK或者Web控制台等服务

兼容性

支持TensorFlow等深度学习框架的标准API，兼容Google CloudML的samples代码，相同模型代码可在不同云平台上训练，避免厂商绑定

高性能

支持超高性能GPU运算，最大可支持56核CPU和128G内存，支持数据并行和模型并行、单机多卡和多机多卡的分布式训练

灵活性

支持按需申请和分配CPU、内存和GPU资源，可根据任务运行时间实现秒级别的计量计费功能

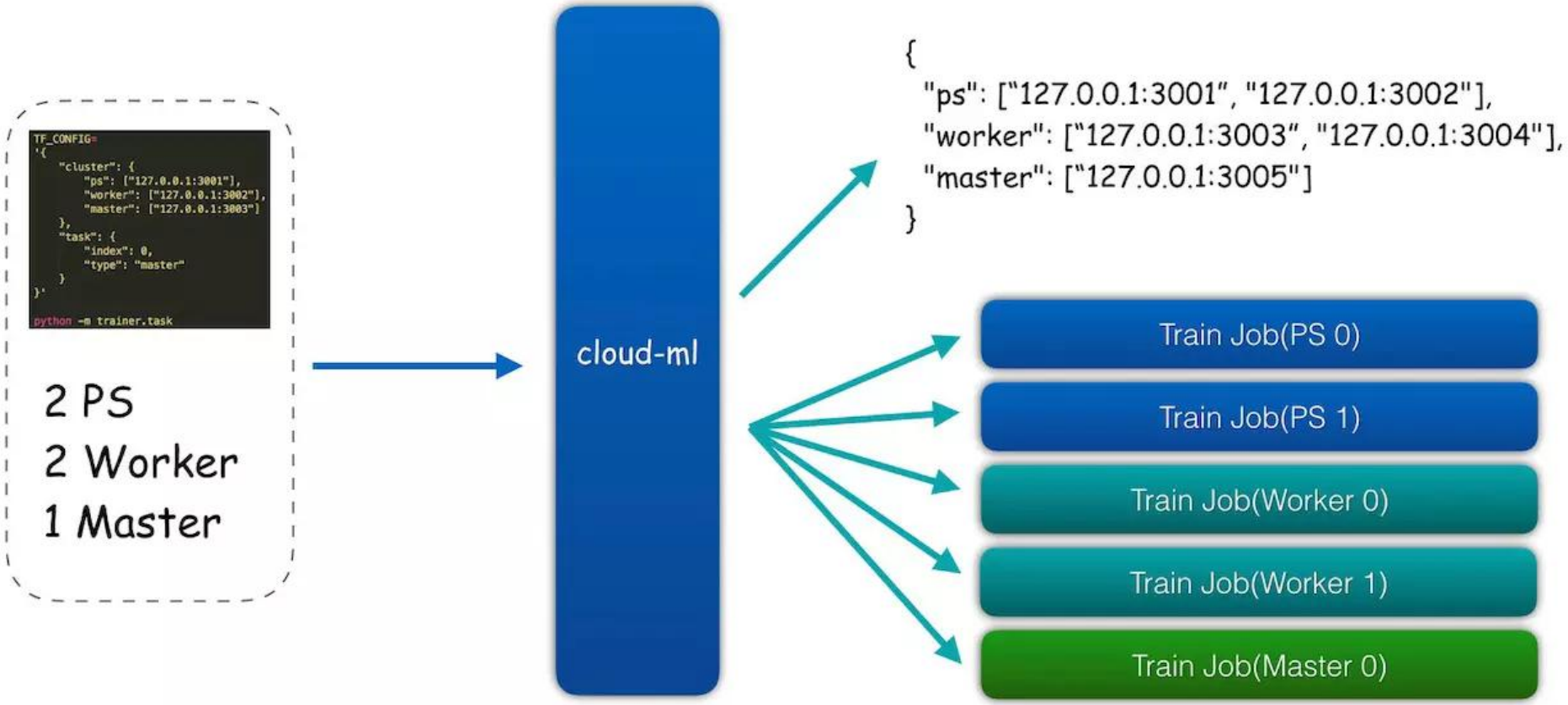
安全性

支持基于Access key/Secret key的多租户认证授权机制，可在线动态调整用户Quota配额

完整性

支持云端训练，支持模型服务，支持开发环境

输入输出


TF_CONFIG=
'{
"cluster": {
"ps": ["127.0.0.1:3001"],
"worker": ["127.0.0.1:3002"],
"master": ["127.0.0.1:3003"]
},
"task": {
"index": 0,
"type": "master"
}
}'
python -m trainer.task
2 PS
2 Worker
1 Master
cloud-ml
{
"ps": ["127.0.0.1:3001", "127.0.0.1:3002"],
"worker": ["127.0.0.1:3003", "127.0.0.1:3004"],
"master": ["127.0.0.1:3005"]
}
Train Job(PS 0)
Train Job(PS 1)
Train Job(Worker 0)
Train Job(Worker 1)
Train Job(Master 0)

总结

小米的cloud-ml主要实现的功能包含

- 顶层通过API、SDK的形式提供深度学习资源调度服务、模型训练服务等，用户只需要通过提交ps/worker/Tensorflow code既可实现分布式深度学习，简单易用
- 在认证和配额上面，cloud-ml自定设计了API Server，采用了AKSK的签名机制，保证认证授权，同时参考OpenStack的多租户和Quata配额机制，实现了Kubernetes上的配额机制
- cloud-ml采用Kubernetes集群实现容器资源/GPU等异构计算资源的调度，同时设计了NodeSelector模块进行GPU粒度调度
- 高可用方面cloud-ml采用了Etcd，能够实现训练过程中出现异常或者错误，能够及时恢复训练任务。同时实现在分布式环境下各个深度学习组件不会出现单点故障

Azure的DSVM

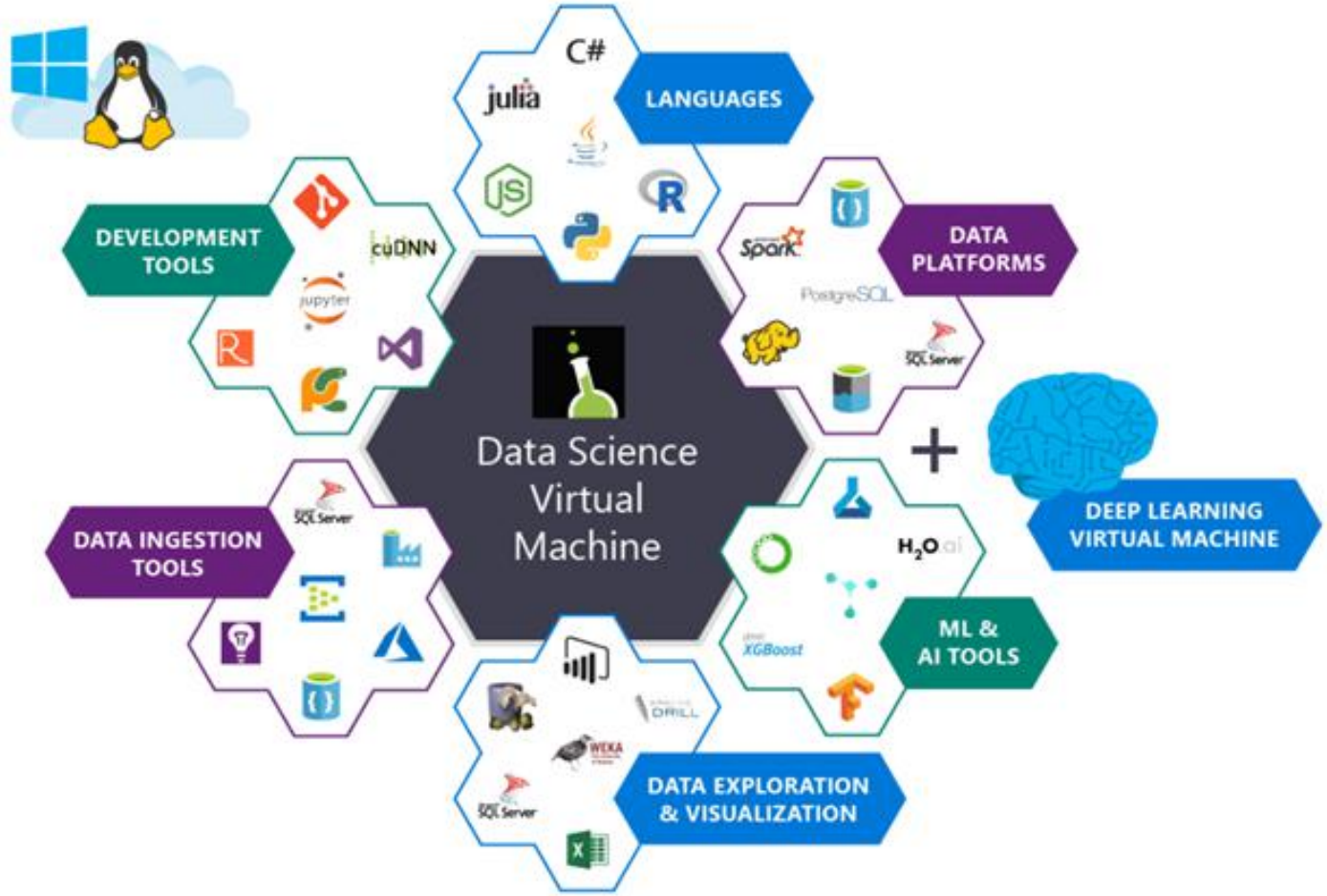
简介

1. DSVM虚拟机全称是 Data Science Virtual Machine，一个基于微软 Azure 云服务定制的虚拟机镜像

2. 内置一系列数据科学和机器学习的开发工具，旨在方便开发者开发和部署机器学习应用软件

3. 主要是针对Caffe进行了打包，同时打包了很多需要使用到的依赖包，用户拉取镜像之后可以一键式部署**机器学习**环境。

整体架构



- 支持多种机器学习工具
- 支持深度学习
- 整个虚拟机打包的环境依赖齐全

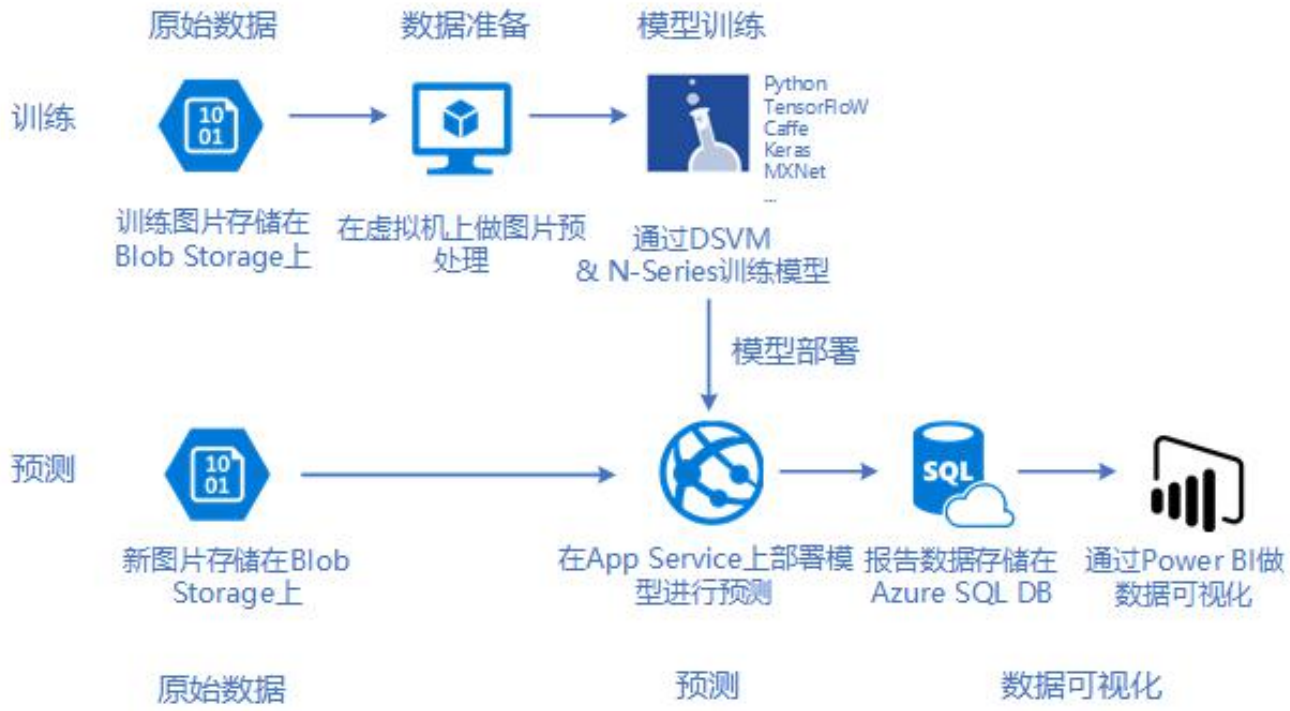
DSVM TO DLVM

1. 深度学习虚拟机是DSVM的特别配置变体
2. 和DSVM具备一样的核心VM镜像
3. 适配绝大部分深度学习框架

   Microsoft 认知工具包(Microsoft Cognitive Toolkit, CNTK)、TensorFlow、Horovod、Keras、Caffe、Caffe2、Chainer、Deep Water、MXNet、NVIDIA DIGITS、Theano、Torch、PyTorch

基于DSVM的深度学习的图像分类方案

- 本实例中的输入输出可以作为一个参考



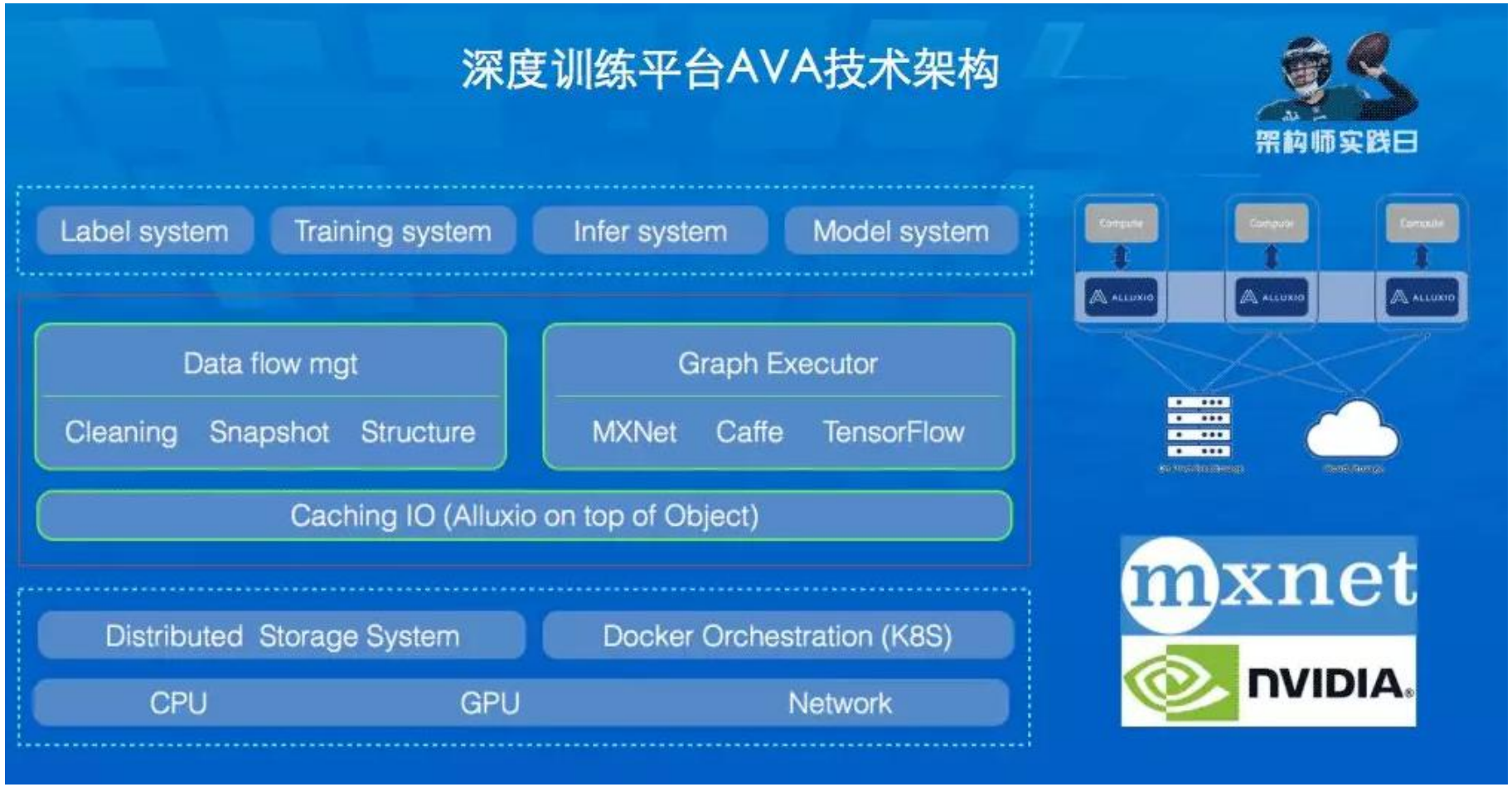
总结

- DSVM是微软Azure上的一个数据科学虚拟机，出发点是一个机器学习平台
- DSVM的一个特殊配置可以直接面向深度学习平台。用户只需要在Azure上面申请DSVM镜像，之后进行对应的资源配置申请，即可启动一个深度学习的环境
- DSVM深度学习环境支持绝大多数的深度学习框架，其中对Caffe的支持最好
- 使用DSVM会极大减少用户进行深度学习开发过程中平台环境部署的时间

七牛云AVA

- 做存储起步
- 面临非法数据识别并过滤问题
- 两个核心功能
  - 数据管理
  - 计算资源管理(GPU)

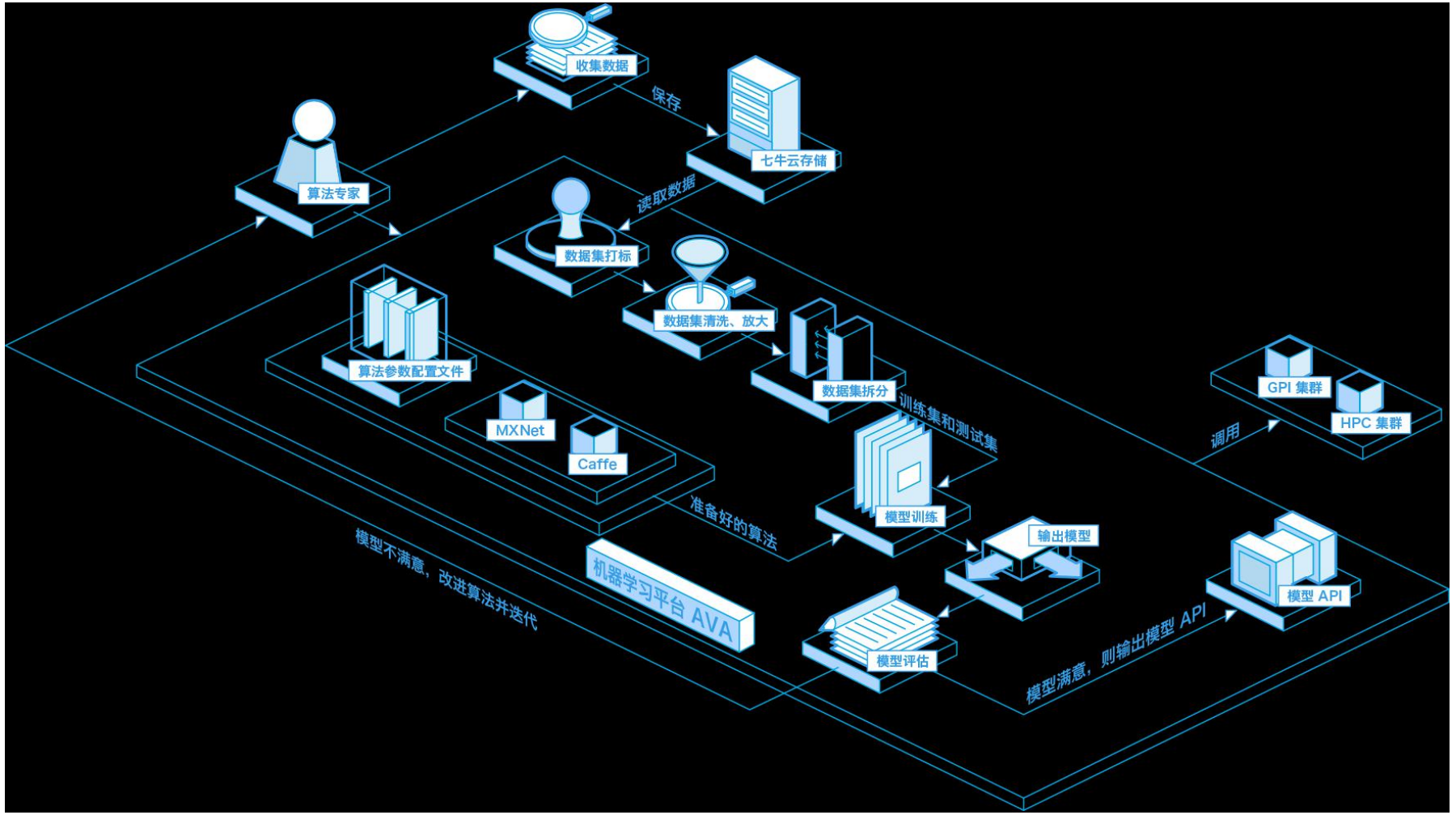
AVA技术架构


深度训练平台AVA技术架构
架构师实践日
Label system
Training system
Infer system
Model system
Data flow mgt
Cleaning
Snapshot
Structure
Graph Executor
MXNet
Caffe
TensorFlow
Caching IO (Alluxio on top of Object)
Distributed Storage System
Docker Orchestration (K8S)
CPU
GPU
Network
ALLUXIO
mxnet
NVIDIA

输入输出流程

- 数据处理，比如图片视频处理：裁减翻转，加水印等。这样可以把数据转换成我们深度训练需要的格式
- 将数据喂到深度训练系统
- 数据喂进来以后，经过深度训练、视频截帧，发现其中的关系，比如判断图片是不是暴力的、恐怖的，再重新把数据放在结构化存储里
- 然后反过来做一个迭代。做一个基础模型，内容分发，重新进入系统做一个循环
- 在循环的不停迭代中，不停改变训练的精度，输出一个最终想要的结果

输入输出流程


收集数据
保存
七牛云存储
算法专家
读取数据
数据集打标
数据集清洗、放大
数据集拆分
算法参数配置文件
MXNet
Caffe
训练集和测试集
调用
GPI 集群
HPC 集群
模型训练
输出模型
模型 API
准备好的算法
模型不满意，改进算法并迭代
机器学习平台 AVA
模型评估
模型满意，则输出模型 API

平台概貌


七牛深度训练平台 - AVA
架构师实践日
图像、视频定制化服务
鉴黄产品
人脸识别
内容识别
目标检测
图像检测
图片检索
视频跟踪
视频检索
数据集管理
数据标注
模型管理
深度训练
联合调度
评估推理

各个平台的核心Feature

- 计算资源调度
- 良好的扩展性
- 兼容性(原生代码兼容)
- 资源共享、资源隔离(主要通过K8s-Docker)
- 支持多种深度学习框架(Tensorflow、Caffe)
- 基于多种分布式存储的统一数据管理
- 可视化界面
- Checkpoint 功能

Tensorflow实践

实现一个简单的线性回归

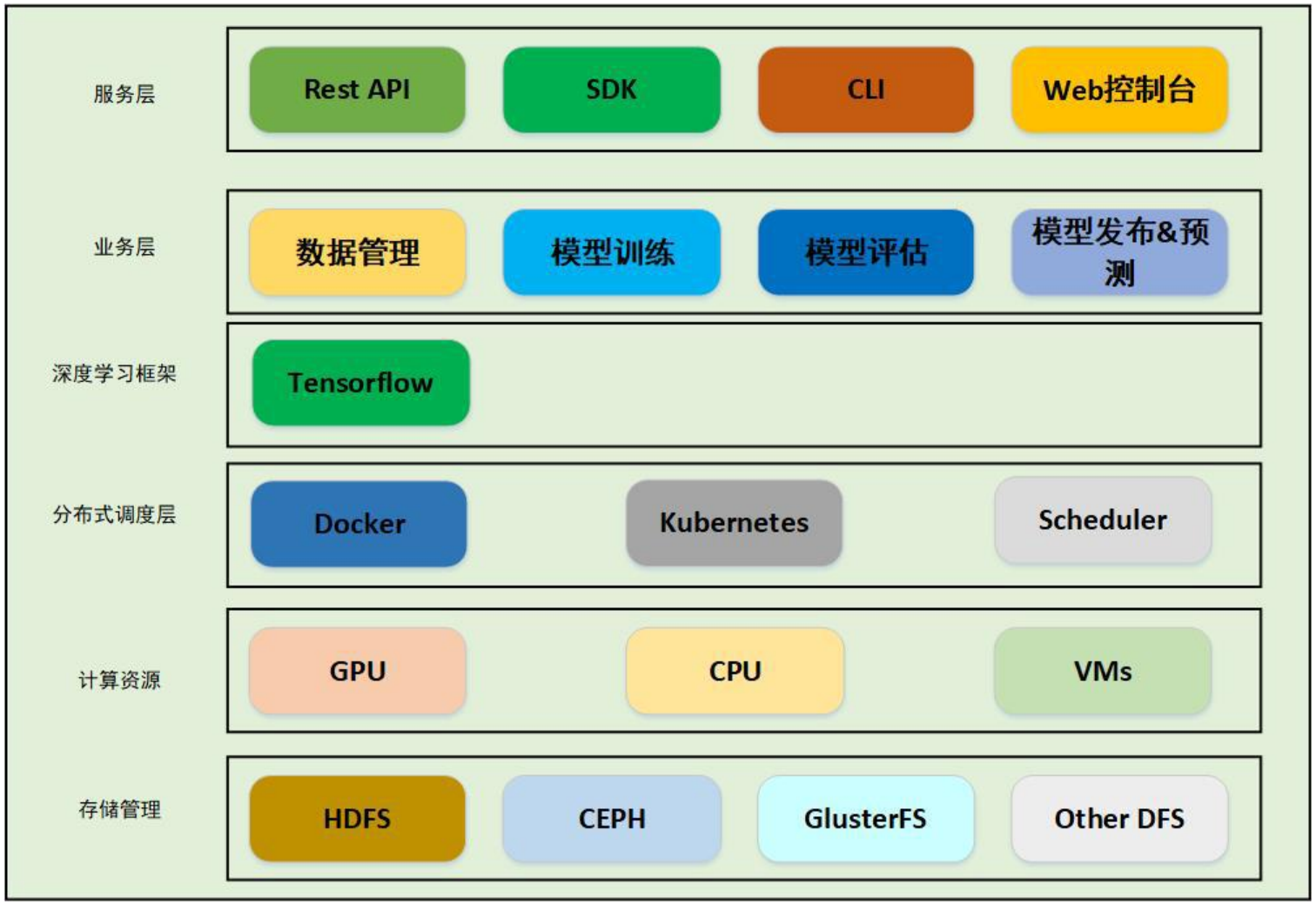
首先用numpy生成一个基于y=2*x + 10线性关系的一组数据，作为训练数据使用

```
Epoch: 1, w: -0.8702505826950073, b: 9.692294120788574
Epoch: 2, w: 0.2823978662490845, b: 10.463948249816895
Epoch: 3, w: 1.0754438638687134, b: 10.317096710205078
Epoch: 4, w: 1.5055475234985352, b: 10.172473907470703
Epoch: 5, w: 1.7277346849441528, b: 10.089864730834961
Epoch: 6, w: 1.8411688804626465, b: 10.046679496765137
Epoch: 7, w: 1.8989087343215942, b: 10.024563789367676
Epoch: 8, w: 1.9282759428024292, b: 10.013301849365234
Epoch: 9, w: 1.9432101249694824, b: 10.007572174072266
Epoch: 10, w: 1.9508051872253418, b: 10.004653930664062
```

求解斜率和截距，以0.01的学习速率最小化交叉熵

```
#!/usr/bin/env python

import numpy as np
import tensorflow as tf

# Prepare train data
train_X = np.linspace(-1, 1, 100)
train_Y = 2 * train_X + np.random.randn(*train_X.shape) * 0.33 + 10

# Define the model
X = tf.placeholder("float")
Y = tf.placeholder("float")
w = tf.Variable(0.0, name="weight")
b = tf.Variable(0.0, name="bias")
loss = tf.square(Y - X * w - b)
train_op = tf.train.GradientDescentOptimizer(0.01).minimize(loss)

# Create session to run
with tf.Session() as sess:
    sess.run(tf.initialize_all_variables())

    epoch = 1
    for i in range(10):
        for (x, y) in zip(train_X, train_Y):
            _, w_value, b_value = sess.run([train_op, w, b],
                                           feed_dict={X: x,
                                                      Y: y})
        print("Epoch: {}, w: {}, b: {}".format(epoch, w_value, b_value))
        epoch += 1
```

对平台的设想


服务层
Rest API
SDK
CLI
Web控制台
业务层
数据管理
模型训练
模型评估
模型发布&预测
深度学习框架
Tensorflow
分布式调度层
Docker
Kubernetes
Scheduler
计算资源
GPU
CPU
VMs
存储管理
HDFS
CEPH
GlusterFS
Other DFS

深度学习平台业务

- 数据集管理
- GPU资源配管理
- 存储资源管理(用户是否可以自己带storage)
- 训练模型管理
- 配额Quota管理
- 日志管理
- 监控管理
- 模型上下线/模型下载

接下来的计划

Plan

- 使用包括ava、华为深度学习平台、elearn等，熟悉深度学习平台使用过程，找出不足
- 构建并使用Tensorflow的集群环境，调研为何Tensorflow的cluster模式不能满足使用
- 进一步熟悉Tensorflow的开发，调研了解深度学习开发工程师从开发到发布的流程