英特尔AI开发者技术研讨会

# 数据分析+AI平台技术及案例研究


龚奇源 (Qiyuan Gong)

Intel 机器学习专家

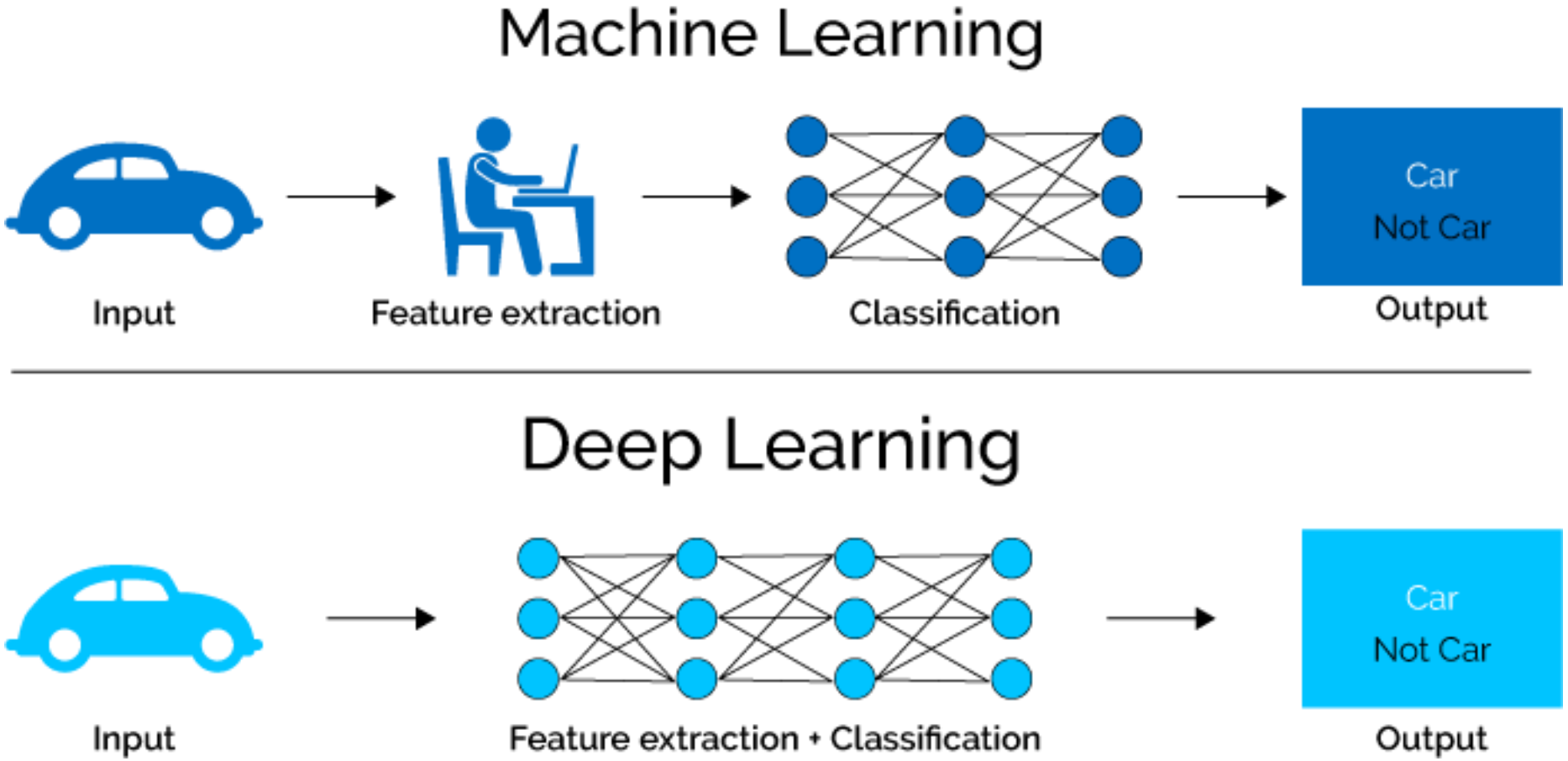
# AGENDA

## Part 1: Intel Analytics-Zoo & Use cases

## Part 2: Cluster Serving with Analytics Zoo

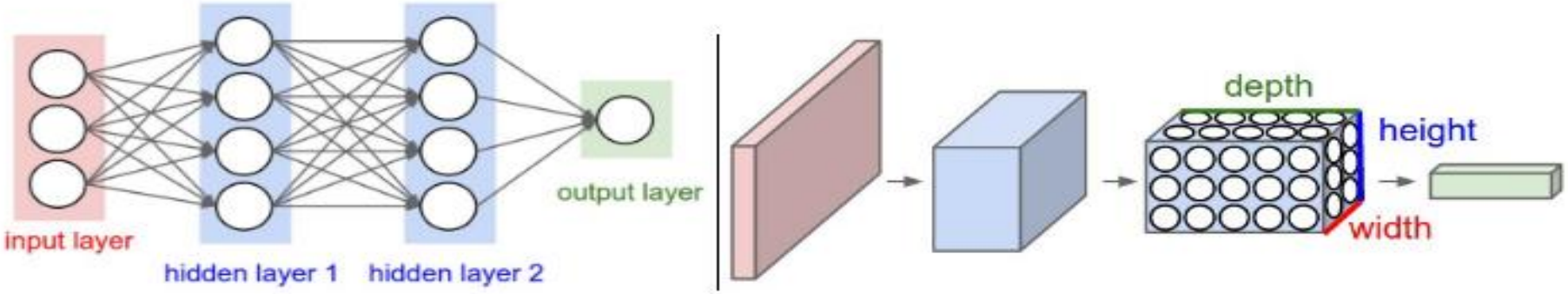

INTEL ANALYTICS-ZOO &
USE CASES

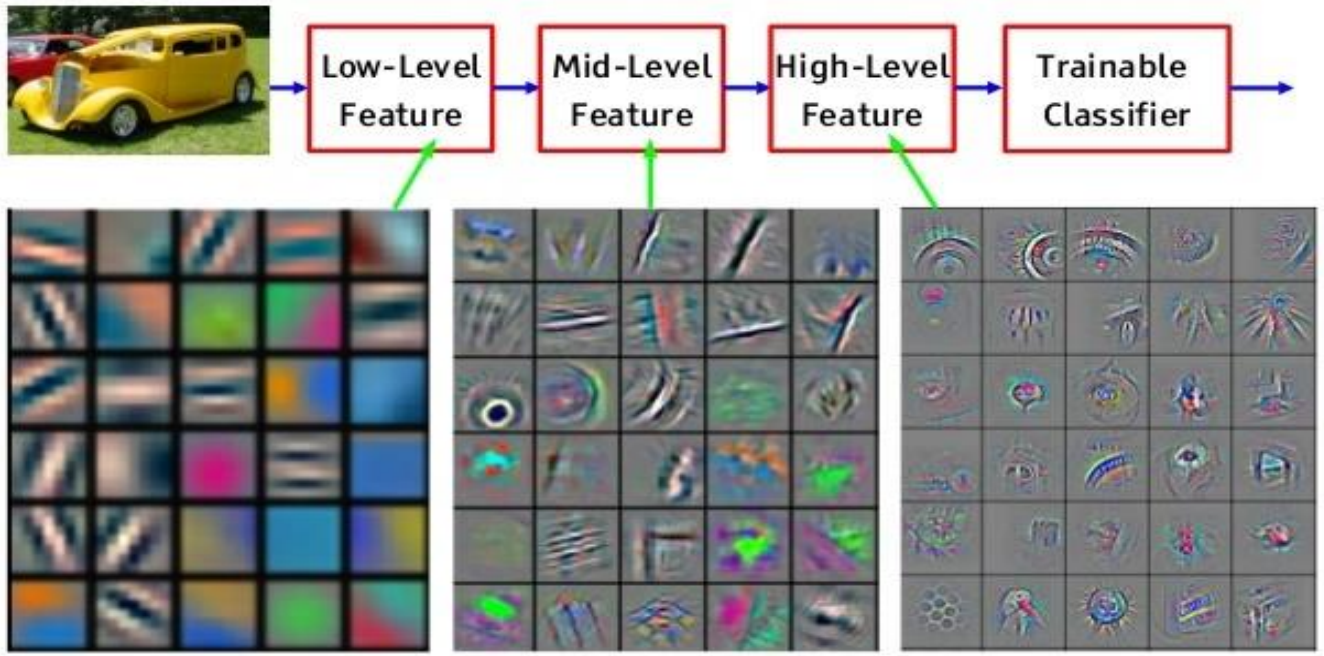
# 深度学习




https://www.quora.com/What-is-the-difference-between-deep-learning-and-usual-machine-learning

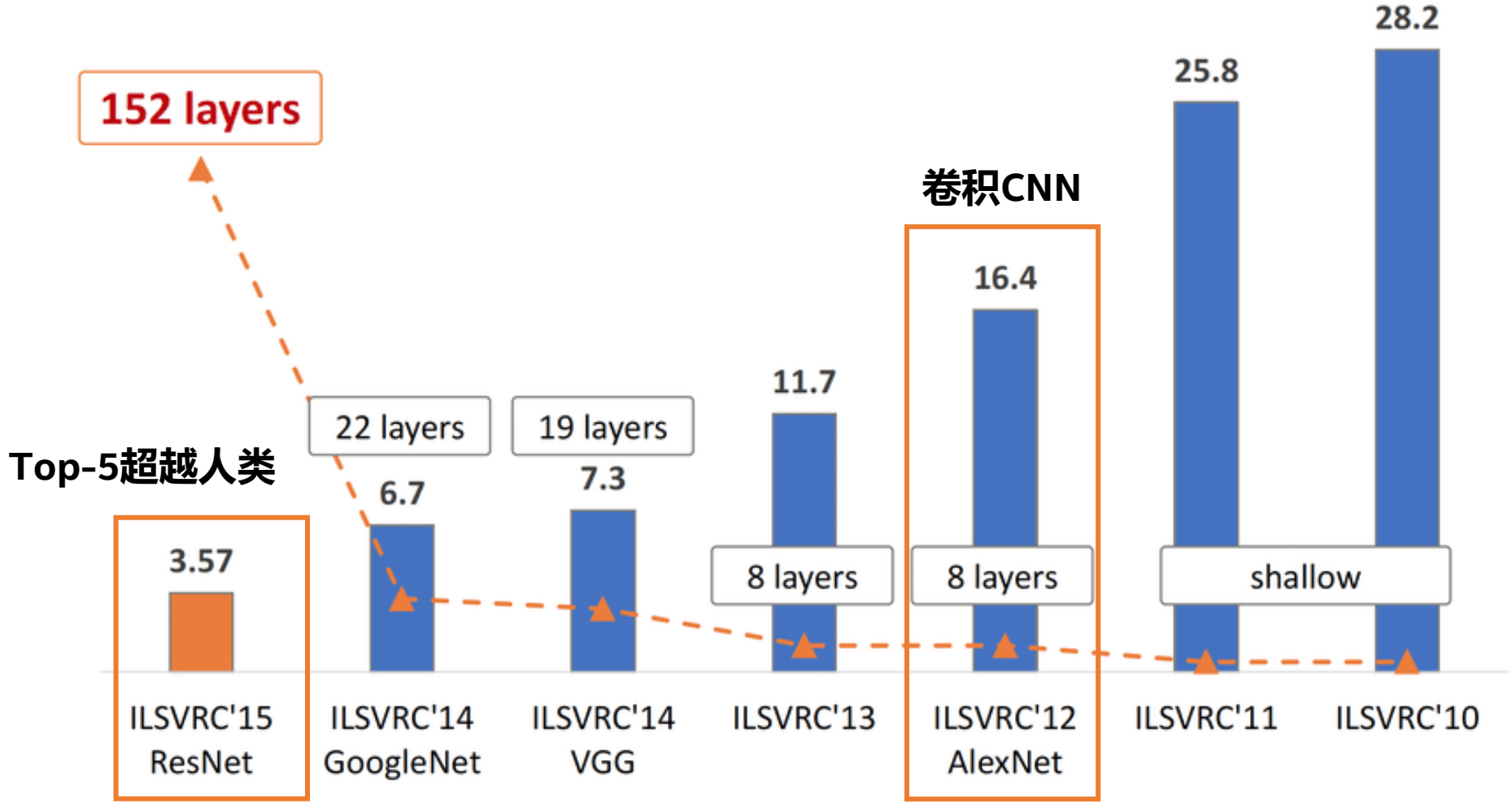
# 深度学习CV




Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

# 深度学习CV




Kaiming etc Deep Residual Learning for Image Recognition,2015

# 深度学习

**2016年 Google Alpha Go 击败前世界冠军李世石**

**In “Nature” 27 January 2016:**

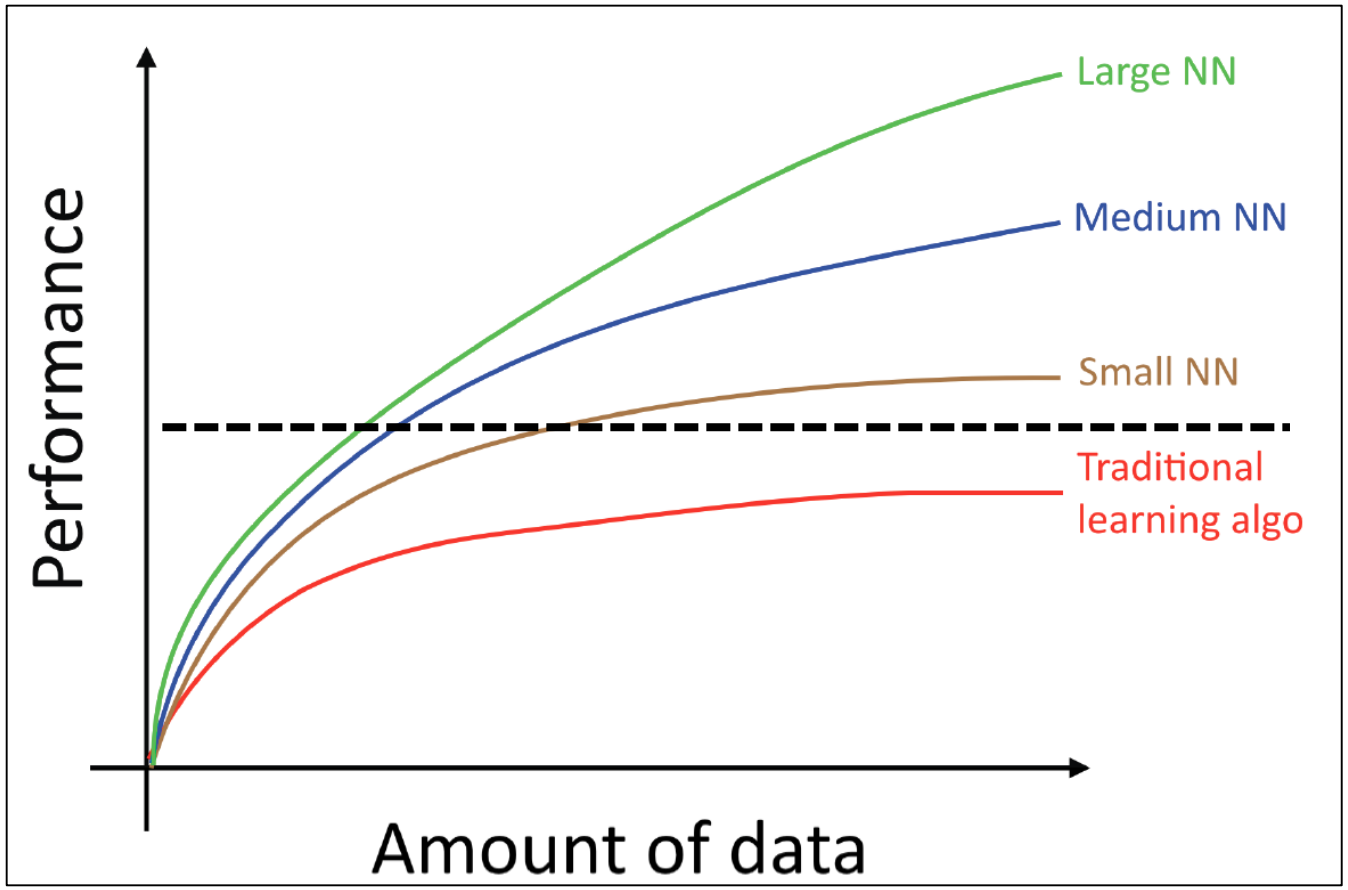
**“DeepMind’s program AlphaGo beat Fan Hui, the European Go champion, five times out of five in tournament conditions...”**

**“...AlphaGo program applied deep learning in neural networks (convolutional NN) — brain-inspired programs in which connections between layers of simulated neurons are strengthened through examples and experience.”**

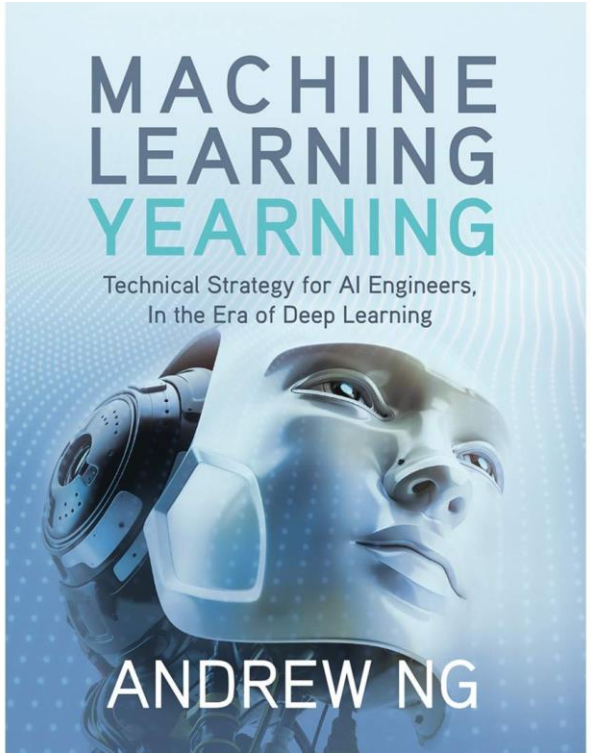
**2017年 Google Alpha Go Master 击败世界冠军柯洁**


https://www.nature.com/nature/volumes/529/issues/7587

# 深度学习




**“Machine Learning Yearning”, Andrew Ng, 2016**

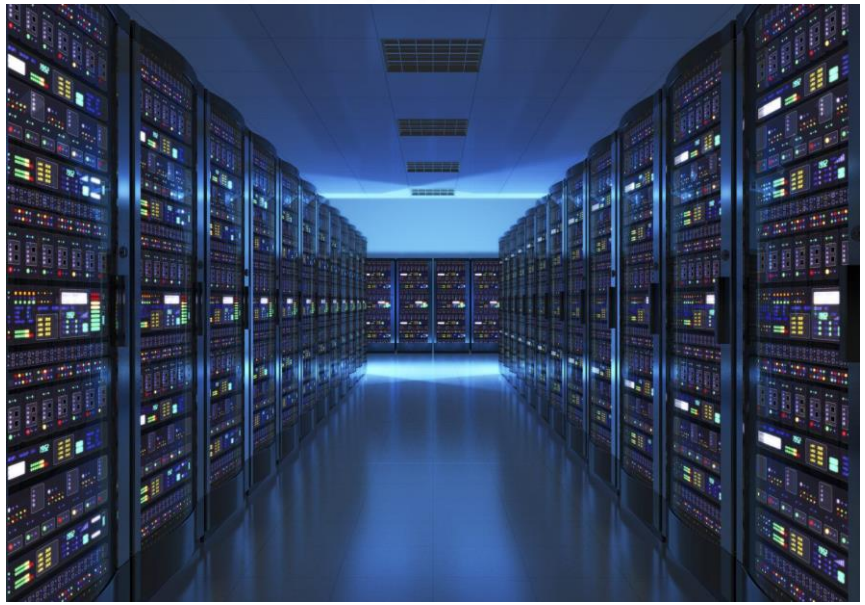


# 深度学习面临的问题

**但天下没有免费的午餐 (no free launch)**

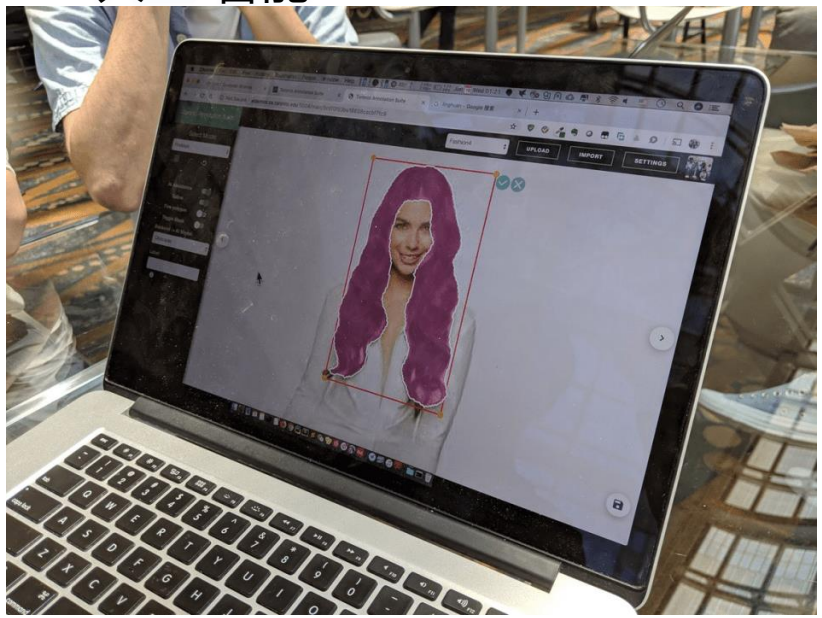
- Deep Learning需要大量**算力 (计算密集)**
- Deep Learning需要大量**数据 (data hungry)**

足够的存储和算力



“人工”智能

IMAGENET



https://medium.com/syncedreview/data-annotation-the-billion-dollar-business-behind-ai-breakthroughs-d929b0a50d23

# 深度学习

**2016年 Google Alpha Go 击败李世石**

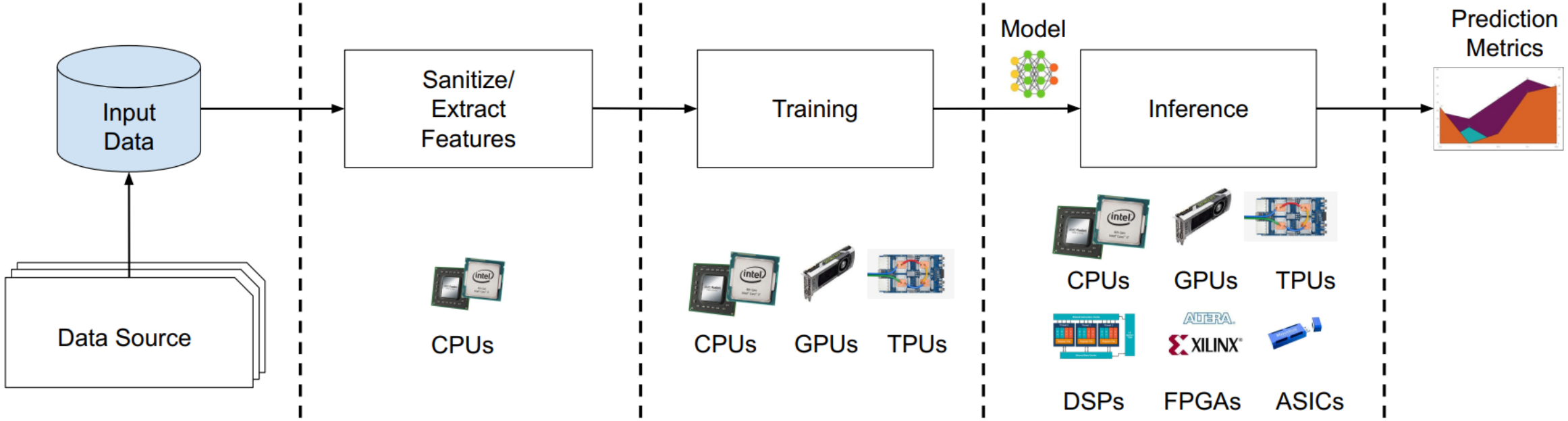
- 现场的Alpha Go所使用的计算资源
  - **48 CPU, 8 GPU ≈ 24 Servers**

- 其实还有一个Distributed Alpha Go
  - **1202 CPU, 176 GPU ≈ 601 Servers**

https://www.nature.com/articles/nature16961

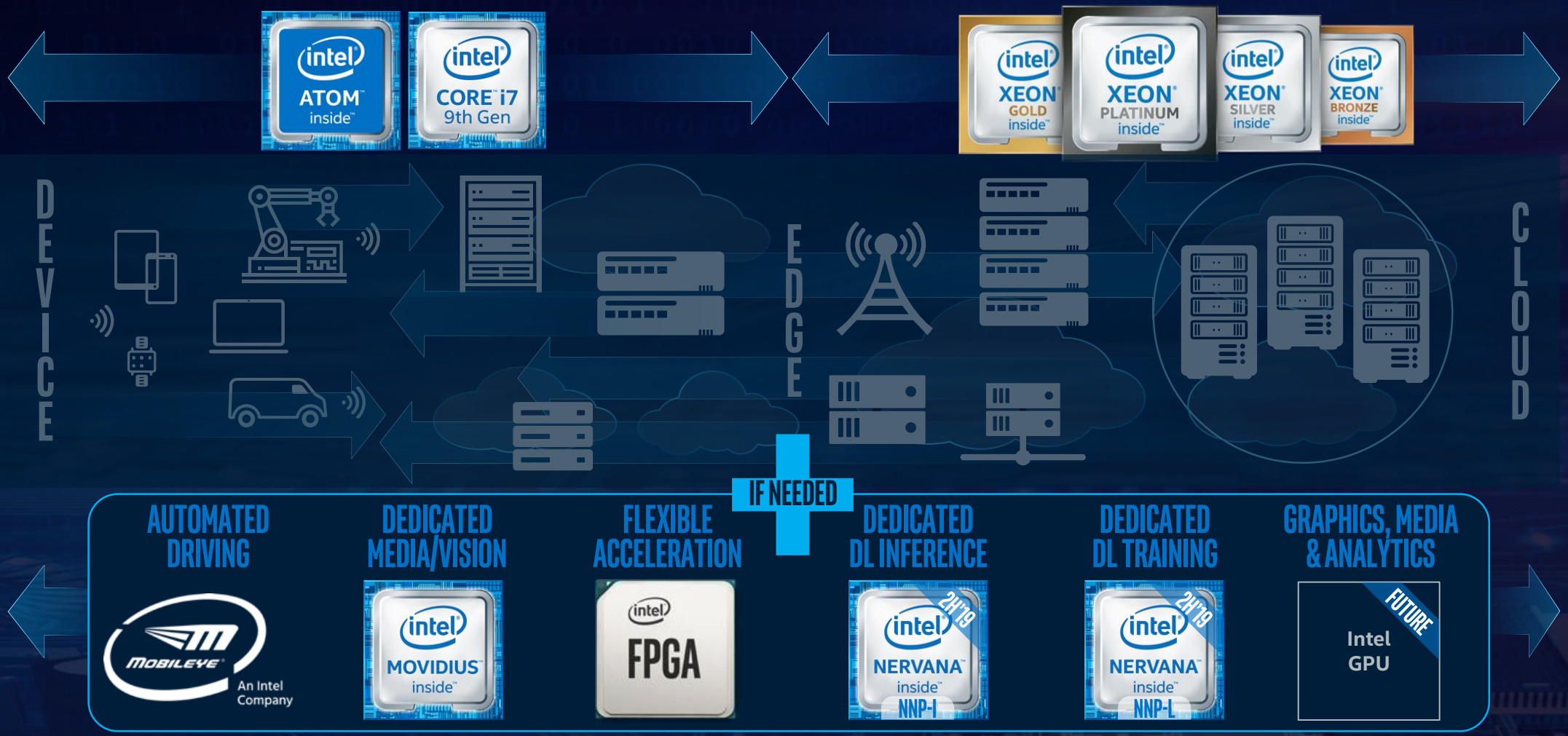
https://newsroom.intel.com/editorials/re-architecting-data-center-intel-xeon-processor-scalable-family/#gs.hi35lo

# 深度学习




https://mlperf.org/
VJ Reddi, 2019, MLPerf Inference Benchmark

# AI at Intel


intel ATOM inside
intel CORE i7 9th Gen
intel XEON GOLD inside
intel XEON PLATINUM inside
intel XEON SILVER inside
intel XEON BRONZE inside
DEVICE
EDGE
CLOUD
IF NEEDED
AUTOMATED DRIVING
MOBILEYE An Intel Company
DEDICATED MEDIA/VISION
intel MOVIDIUS inside
FLEXIBLE ACCELERATION
intel FPGA
DEDICATED DL INFERENCE
2H'19
intel NERVANA inside NNP-I
DEDICATED DL TRAINING
2H'19
intel NERVANA inside NNP-L
GRAPHICS, MEDIA & ANALYTICS
FUTURE
Intel GPU

# Speed Up Development

## Using Open AI Software

MACHINE LEARNING ⟷ DEEP LEARNING

### TOOLKITS
App developers

**ANALYTICS ZOO**
Open source platform for building E2E Analytics & AI applications on Apache Spark* with distributed TensorFlow*, Keras*, BigDL

**OpenVINO™**
Deep learning inference deployment on CPU/GPU/FPGA/VPU for Caffe*, TensorFlow*, MXNet*, ONNX*, Kaldi*

**NAUTA**
Open source, scalable, and extensible distributed deep learning platform built on Kubernetes (BETA)

### LIBRARIES
Data scientists

**Python**
- Scikit-learn
- Pandas
- NumPy

**R**
- Cart
- Random Forest
- e1071

**Distributed**
- MLlib (on Spark)
- Mahout

**Intel-optimized Frameworks**

TensorFlow* · Caffe2* · ONNX* · mxnet* · PyTorch* · BigDL

And more framework optimizations underway including PaddlePaddle*, Chainer*, CNTK* & others

### KERNELS
Library developers

**Intel® Distribution for Python***
*Intel distribution optimized for machine learning*

**Intel® Data Analytics Acceleration Library (DAAL)**
*High performance machine learning & data analytics library*

**Intel® Math Kernel Library for Deep Neural Networks (MKL-DNN)**
*Open source DNN functions for CPU / integrated graphics*

**nGraph**
*Open source compiler for deep learning model computations optimized for multiple devices (CPU, GPU, NNP) from multiple frameworks (TF, MXNet, ONNX)*

# 以数据为中心的世界

全球超过 OVER

一半 HALF 数据 OF THE WORLD'S DATA

创建于过去

WAS CREATED IN THE LAST

两年 2 YEARS

其中只有不到 LESS THAN

2% 的数据 HAS BEEN ANALYZED

经过了分析

# 大规模人工智能应用

## 正面临巨大的挑战

- 复杂性
- 成本
- 可扩展性
- 专有接口
- 数据隐私

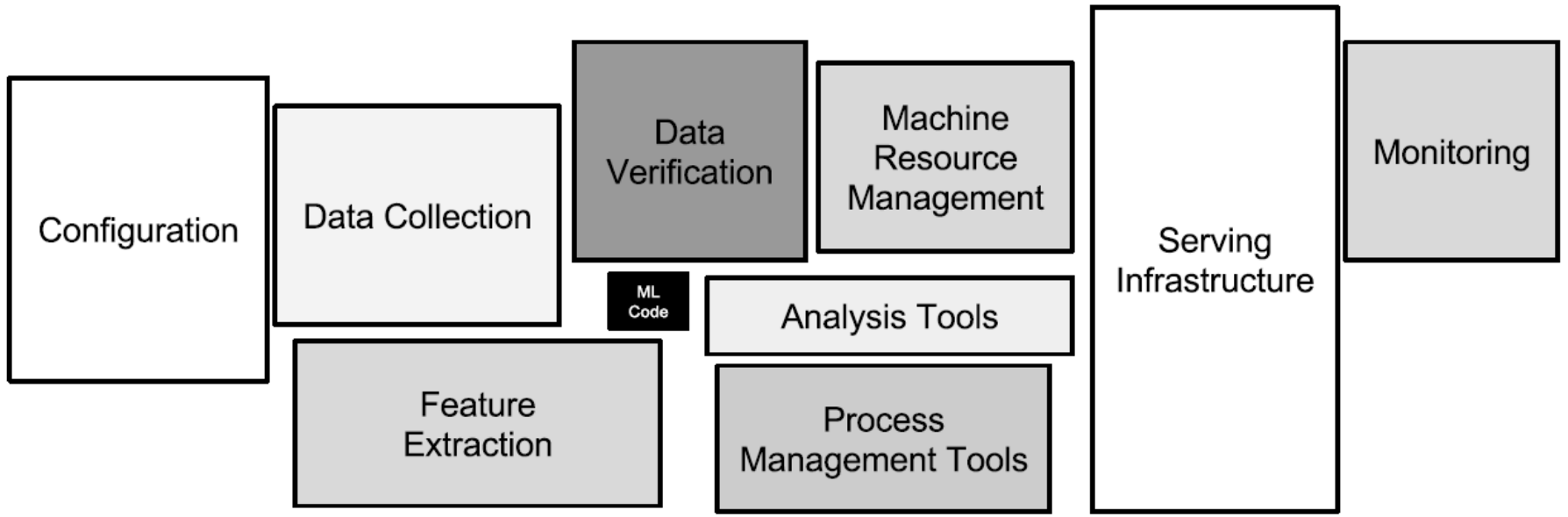



Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

"Hidden Technical Debt in Machine Learning Systems",
Sculley et al., Google, NIPS 2015

# 统一的数据分析及AI

获取 / 存储 | 清洗 / 准备 | 分析 / 建模 | 部署 / 可视化

集成的数据流水线


数据管理 | 数据分析 | 数据科学及人工智能

# 大数据上的人工智能

BigDL

基于 Apache Spark 的高性能
深度学习框架

software.intel.com/bigdl

ANALYTICS ZOO

统一的分析+人工智能平台

基于 Apache Spark 的分布式
TensorFlow、Keras 和 BigDL

参考用例、人工智能模型、高级API、特征工程等

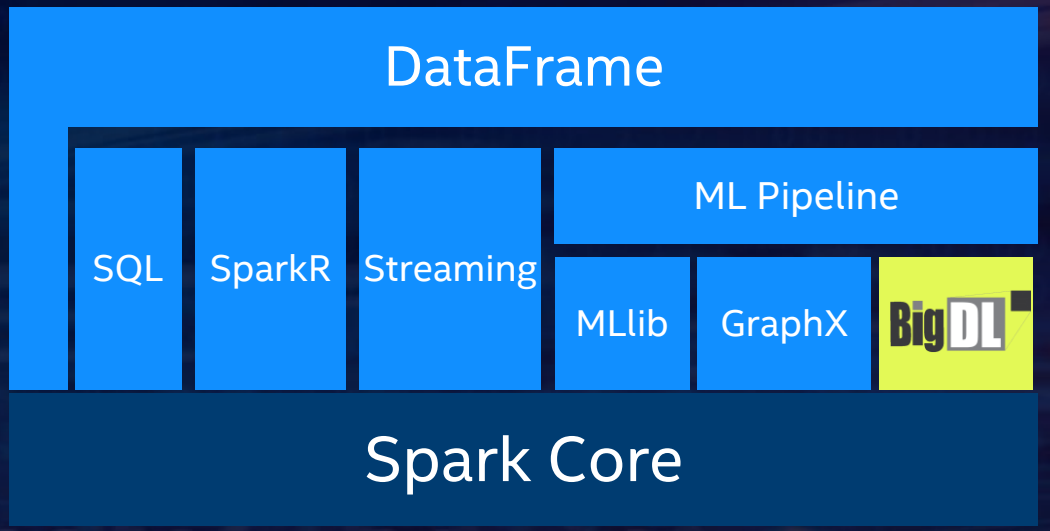
https://github.com/intel-analytics/analytics-zoo

## 加快数据分析及人工智能大规模应用

# BigDL

## Bringing Deep Learning To Big Data Platform

- Distributed deep learning framework for Apache Spark
- Make deep learning more accessible to big data users and data scientists
  - Write deep learning applications as ***standard Spark programs***
  - Run on existing Spark/Hadoop clusters (***no changes needed***)
- Feature parity with popular deep learning frameworks
  - E.g., Caffe, Torch, Tensorflow, etc.
- High performance (on CPU)
  - Powered by Intel MKL and multi-threaded programming
- Efficient scale-out
  - Leveraging Spark for distributed training & inference




https://github.com/intel-analytics/BigDL

https://bigdl-project.github.io/

# BigDL Run as Standard Spark Programs

## Standard Spark jobs

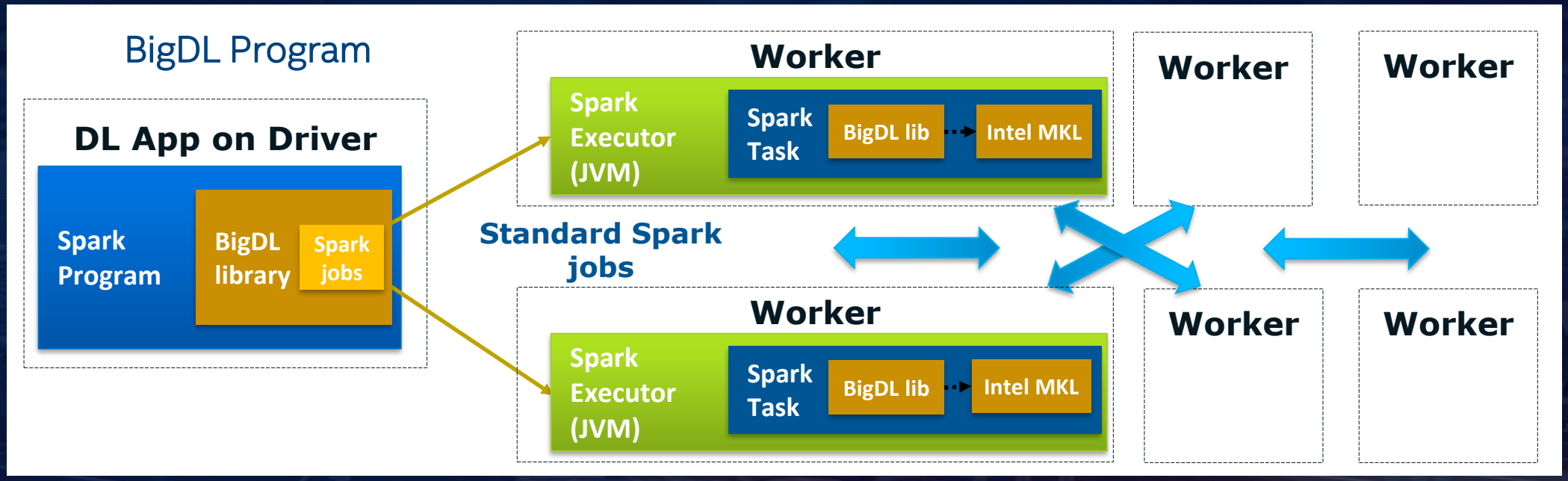
- No changes to the Spark or Hadoop clusters needed

## Iterative

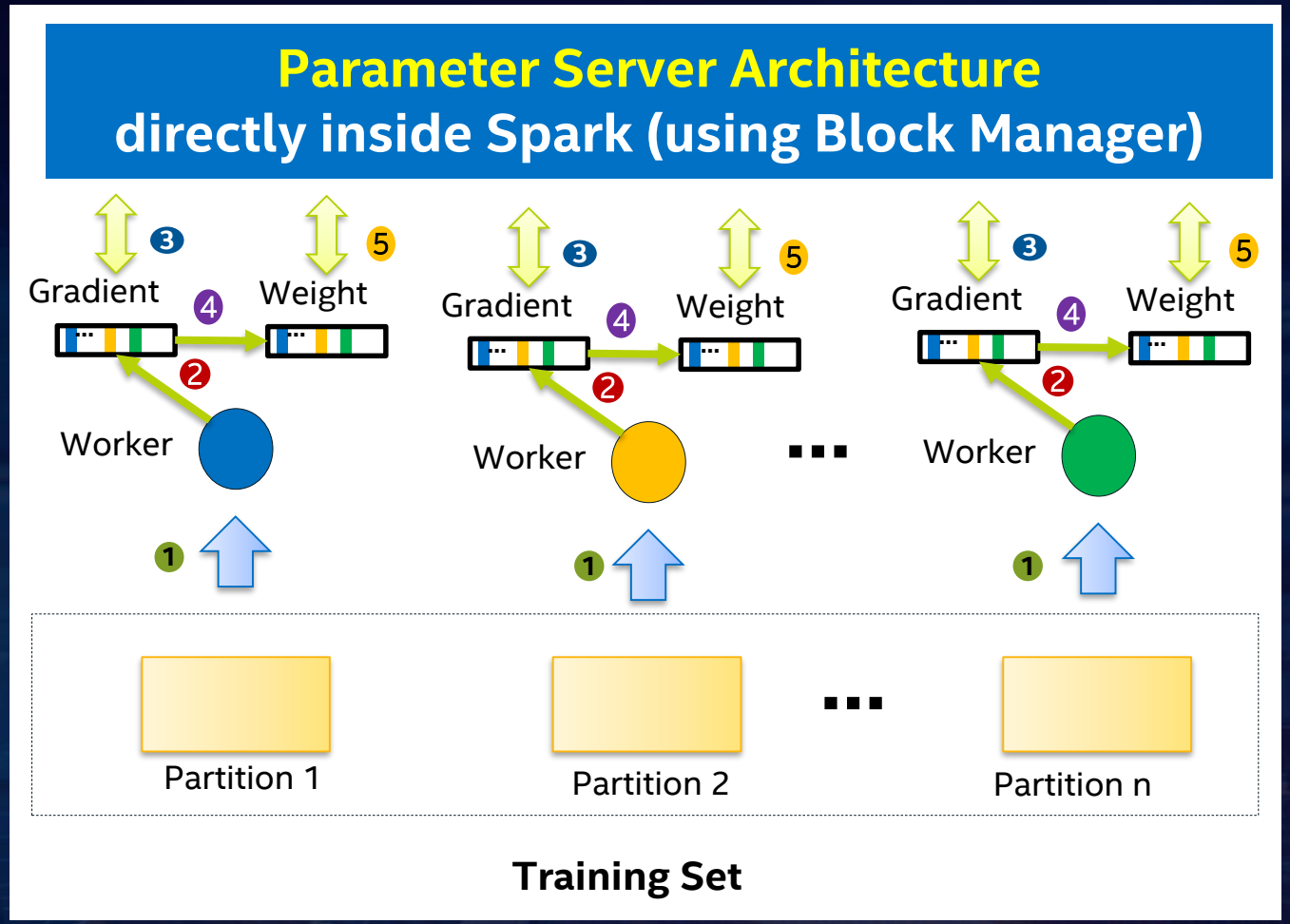
- Each iteration of the training runs as a Spark job

## Data parallel

- Each Spark task runs the same model on a subset of the data (batch)



# Distributed Training in BigDL




Peer-2-Peer **All-Reduce** synchronization

# BigDL: A Distributed Deep Learning Framework for Big Data

Jason Dai, Yiheng Wang, Xin Qiu, Ding Ding, Yao Zhang, Yanzhang Wang, Xianyan Jia, Cherry Zhang, Yan Wan, Zhichao Li, Jiao Wang, Shengsheng Huang, Zhongyuan Wu, Yang Wang, Yuhao Yang, Bowen She, Dongjie Shi, Qi Lu, Kai Huang, Guoqiong Song

*(Submitted on 16 Apr 2018 (v1), last revised 5 Nov 2019 (this version, v4))*

This paper presents BigDL (a distributed deep learning framework for Apache Spark), which has been used by a variety of users in the industry for building deep learning applications on production big data platforms. It allows deep learning applications to run on the Apache Hadoop/Spark cluster so as to directly process the production data, and as a part of the end-to-end data analysis pipeline for deployment and management. Unlike existing deep learning frameworks, BigDL implements distributed, data parallel training directly on top of the functional compute model (with copy-on-write and coarse-grained operations) of Spark. We also share real-world experience and "war stories" of users that have adopted BigDL to address their challenges(i.e., how to easily build end-to-end data analysis and deep learning pipelines for their production data).

Comments: In ACM Symposium of Cloud Computing conference (SoCC) 2019
Subjects: **Distributed, Parallel, and Cluster Computing (cs.DC)**; Artificial Intelligence (cs.AI); Machine Learning (cs.LG)
DOI: 10.1145/1122445.1122456
Cite as: arXiv:1804.05839 [cs.DC]
(or arXiv:1804.05839v4 [cs.DC] for this version)

ACM Symposium on Cloud Computing

11:00 AM
12:30 PM

Session 2 (Systems for ML and Data Mining)
Chair: Neeraja Yadwadkar

BigDL: A Distributed Deep Learning Framework for Big Data
Jinquan Dai (Intel), Yiheng Wang (Tencent), Xin Qiu (Intel), Ding Ding (Intel), Yao Zhang (Sequoia Capital), Yanzhang Wang (Intel), Xianyan Jia (Alibaba), Li Zhang (Intel), Yan Wan (Alibaba), Zhichao Li (Intel), Jiao Wang (Intel), Shengsheng Huang (Intel), Zhongyuan Wu (Intel), Yang Wang (Intel), Yuhao Yang (Intel), Bowen She (Intel), Dongjie Shi (Intel), Qi Lu (Intel), Kai Huang (Intel), Guoqiong Song (Intel)

## ACM Symposium on Cloud Computing 2019

Chaminade, Santa Cruz, California

November 20-23, 2019

https://arxiv.org/abs/1804.05839
https://acmsocc.github.io/2019/schedule.html

# Analytics Zoo

## 统一的大数据分析+人工智能平台

# 统一的数据分析和AI流水线

## 端到端、从原型到生产化部署的无缝扩展

在笔记本电脑上使用样本数据构建原型

在集群上使用历史数据运行模型试验

在分布式生产环境中部署


- 从笔记本电脑到分布式集群几乎**无需任何代码更改**
- 无需数据拷贝，**直接访问生产大数据系统**
- 高效构建**端到端的**数据分析+ AI 流水线原型
- 无缝扩展部署到**大数据集群及生产环境**

# Analytics Zoo

## 统一的大数据分析+人工智能平台

**用户案例**

Recommendation | Anomaly Detection | Text Classification | Text Matching

**模型**

Image Classification | Object Detection | Seq2Seq | Transformer | BERT

**特征工程**

image | 3D image | text | time series

**高级流水线**

tfpark: Distributed TensorFlow on Spark | Distributed Keras w/ autograd on Spark

nnframes: Spark Dataframes & ML Pipelines for Deep Learning | Distributed Model Inference (batch, streaming & online)

**后端**

TensorFlow* | Keras* | PyTorch* | BigDL | NLP Architect | Apache Spark* | Apache Flink*

Ray* | MKLDNN | OpenVINO | Intel® Optane™ DCPMM | DL Boost (VNNI)

# 基于SPARK的分布式TENSORFLOW流水线

- **Data wrangling and analysis using PySpark**
- **Deep learning model development using TensorFlow or Keras**
- **Distributed training / inference on Spark**

```
#pyspark code
train_rdd = spark.hadoopFile(…).map(…)
dataset = TFDataset.from_rdd(train_rdd,…)

#tensorflow code
import tensorflow as tf
slim = tf.contrib.slim
images, labels = dataset.tensors
with slim.arg_scope(lenet.lenet_arg_scope()):
   logits, end_points = lenet.lenet(images, …)
loss = tf.reduce_mean( \
   tf.losses.sparse_softmax_cross_entropy( \
   logits=logits, labels=labels))

#distributed training on Spark
optimizer = TFOptimizer.from_loss(loss, Adam(…))
optimizer.optimize(end_trigger=MaxEpoch(5))
```

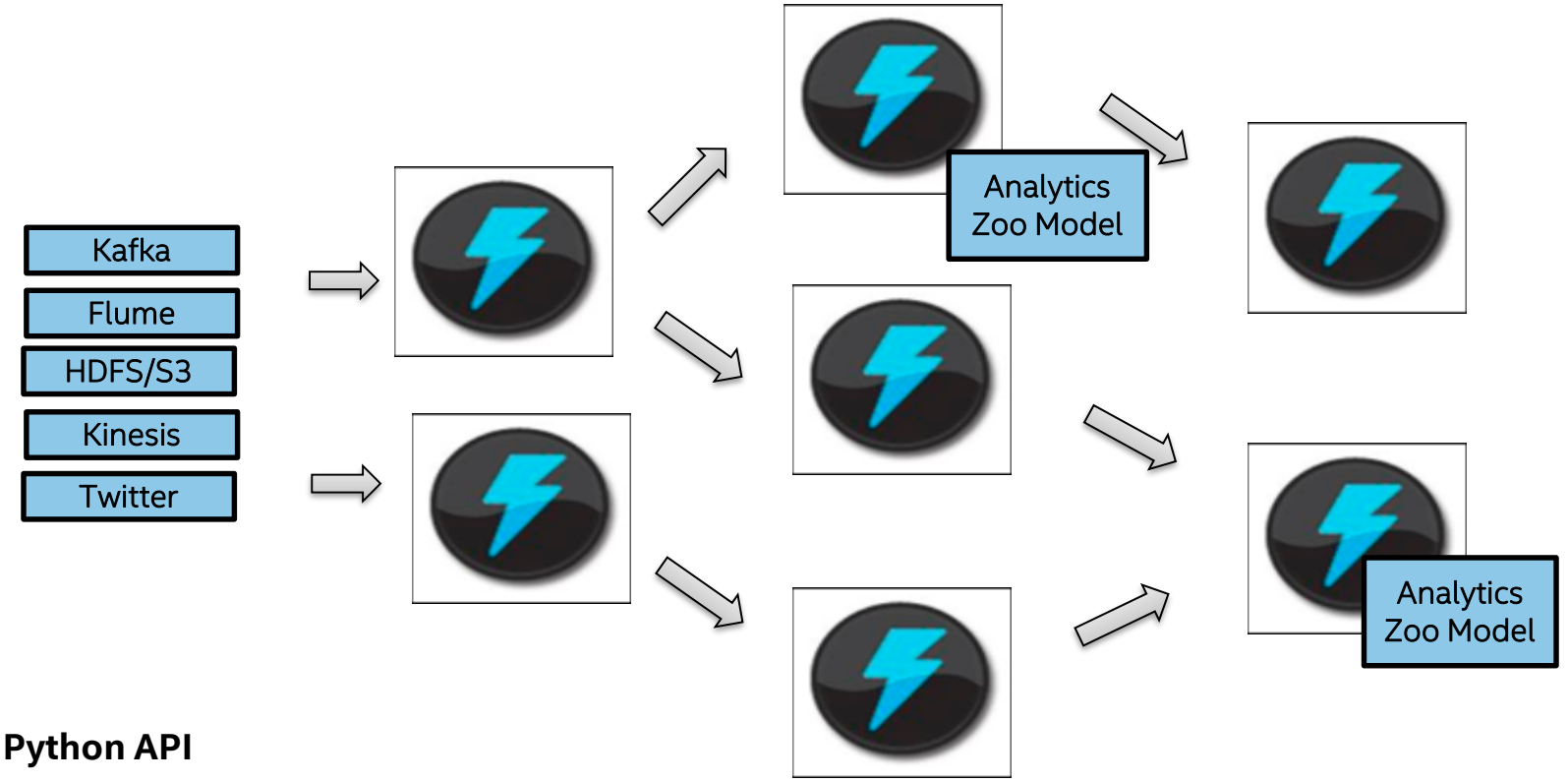
在 PySpark 程序内嵌 TensorFlow 代码

# 基于SPARK DATAFRAME & ML 流水线的深度学习

```
#Spark dataframe transformations
parquetfile = spark.read.parquet(…)
train_df = parquetfile.withColumn(…)

#Keras API
model = Sequential()
          .add(Convolution2D(32, 3, 3, activation='relu', input_shape=…)) \
          .add(MaxPooling2D(pool_size=(2, 2))) \
          .add(Flatten()).add(Dense(10, activation='softmax')))

#Spark ML pipeline
Estimater = NNEstimater(model, CrossEntropyCriterion()) \
                .setLearningRate(0.003).setBatchSize(40).setMaxEpoch(5) \
                .setFeaturesCol("image")
nnModel = estimater.fit(train_df)
```

在 **Spark Dataframe** 和 **ML** 流水线中，直接支持深度神经网络模型

# 分布式、实时(流式)模型推理流水线




- 纯 Java 或 Python API
- 支持 Flink、Kafka、Storm、 Web Service 等
- OpenVINO 和 DL Boost (VNNI) 加速

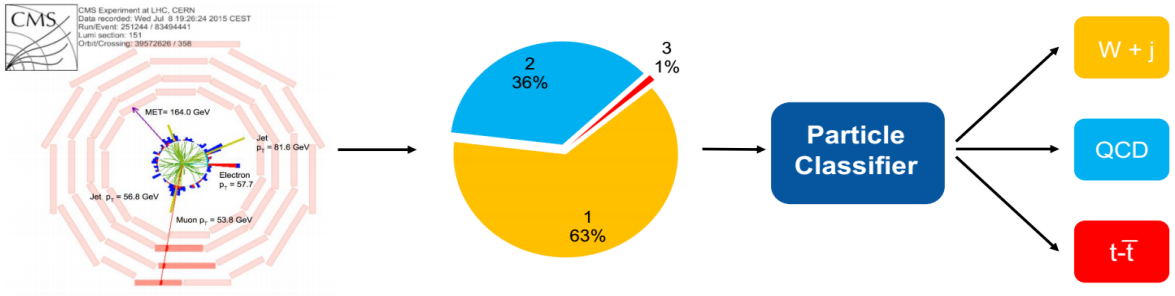
# 跨行业的端到端客户案例


* 文中涉及的其它名称及商标属于各自所有者资产 。

# 阿里APACHE FLINK极客挑战赛

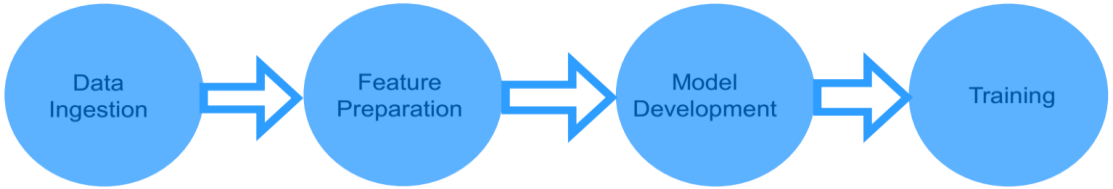

## 这是什么（垃圾）？ 100个类别

# CERN*基于深度学习的高能物理粒子事件分类




高能物理数据的深度学习流水线

## Data Pipeline




Read physics data and feature engineering

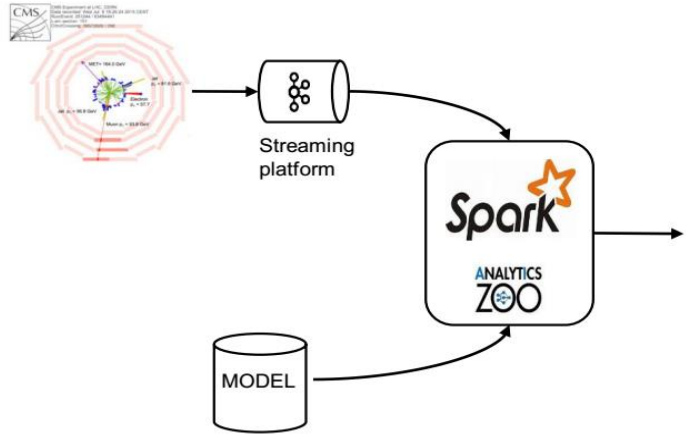
Prepare input for Deep Learning network

1. Specify model topology
2. Tune model topology on small dataset

Train the best model

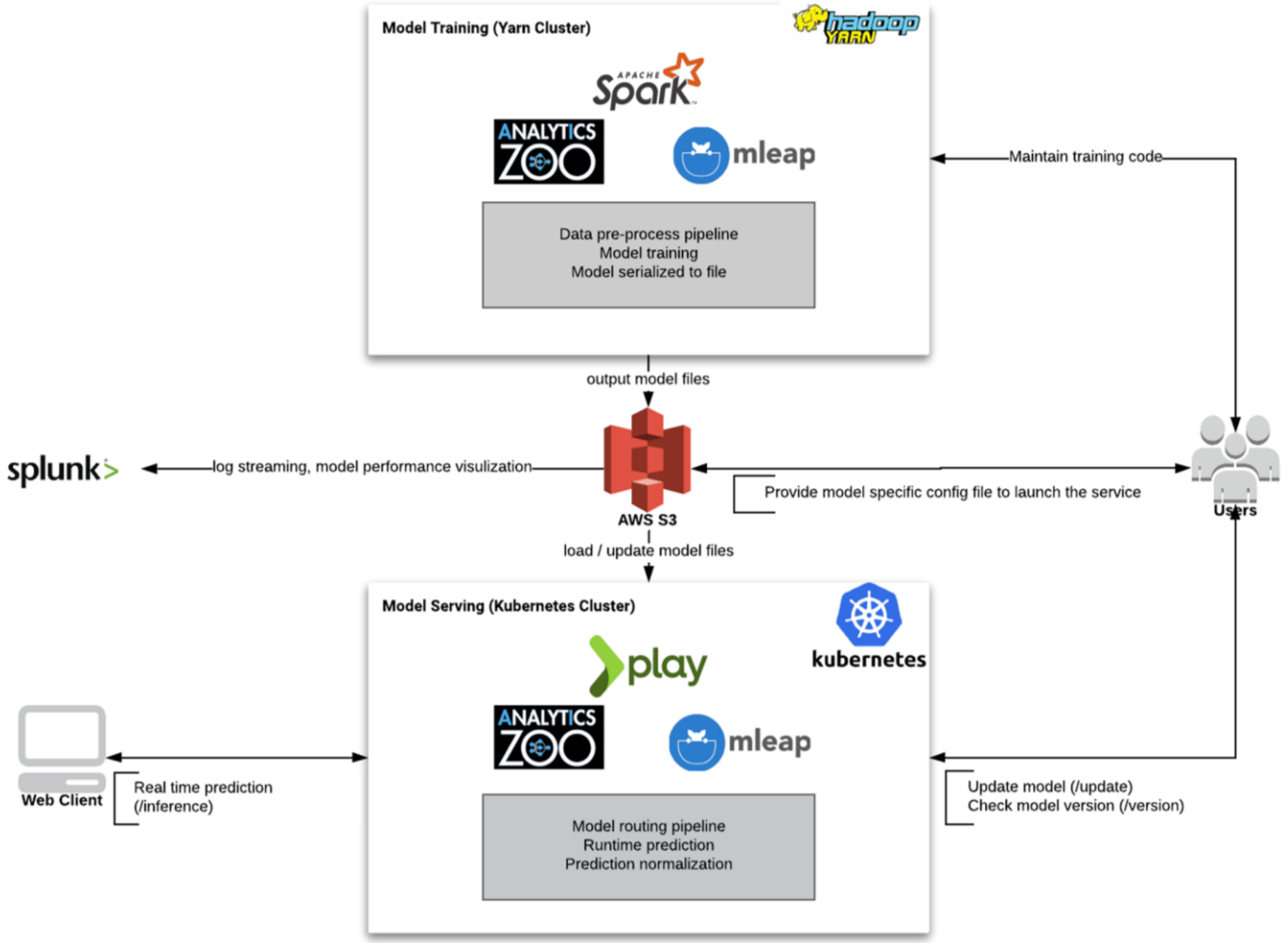
**Leveraging Apache Spark and Analytics Zoo in Python Notebooks**

使用 Apache Kafka 和 Spark 构建模型服务



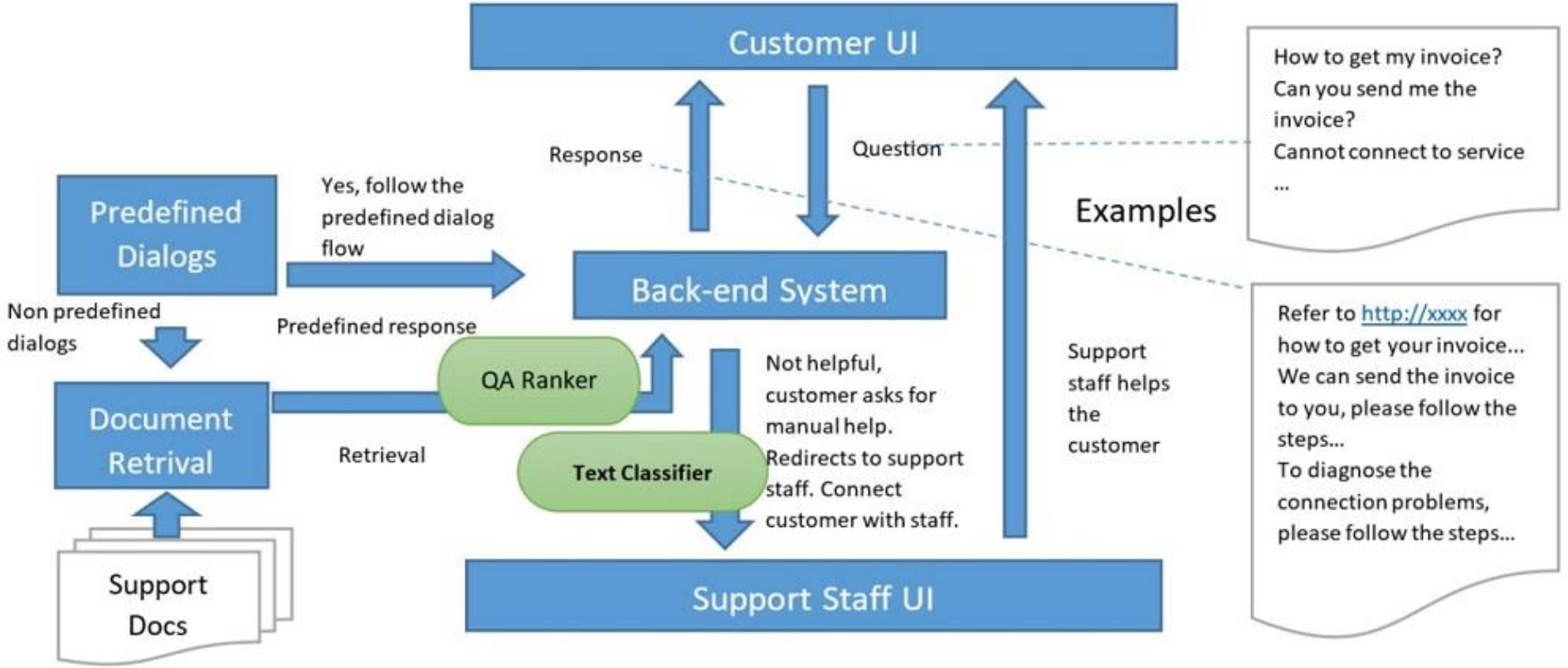

https://db-blog.web.cern.ch/blog/luca-canali/machine-learning-pipelines-high-energy-physics-using-apache-spark-bigdl
https://databricks.com/session/deep-learning-on-apache-spark-at-cerns-large-hadron-collider-with-intel-technologies

# OFFICE DEPOT*： 基于用户 SESSION 行为的产品推荐



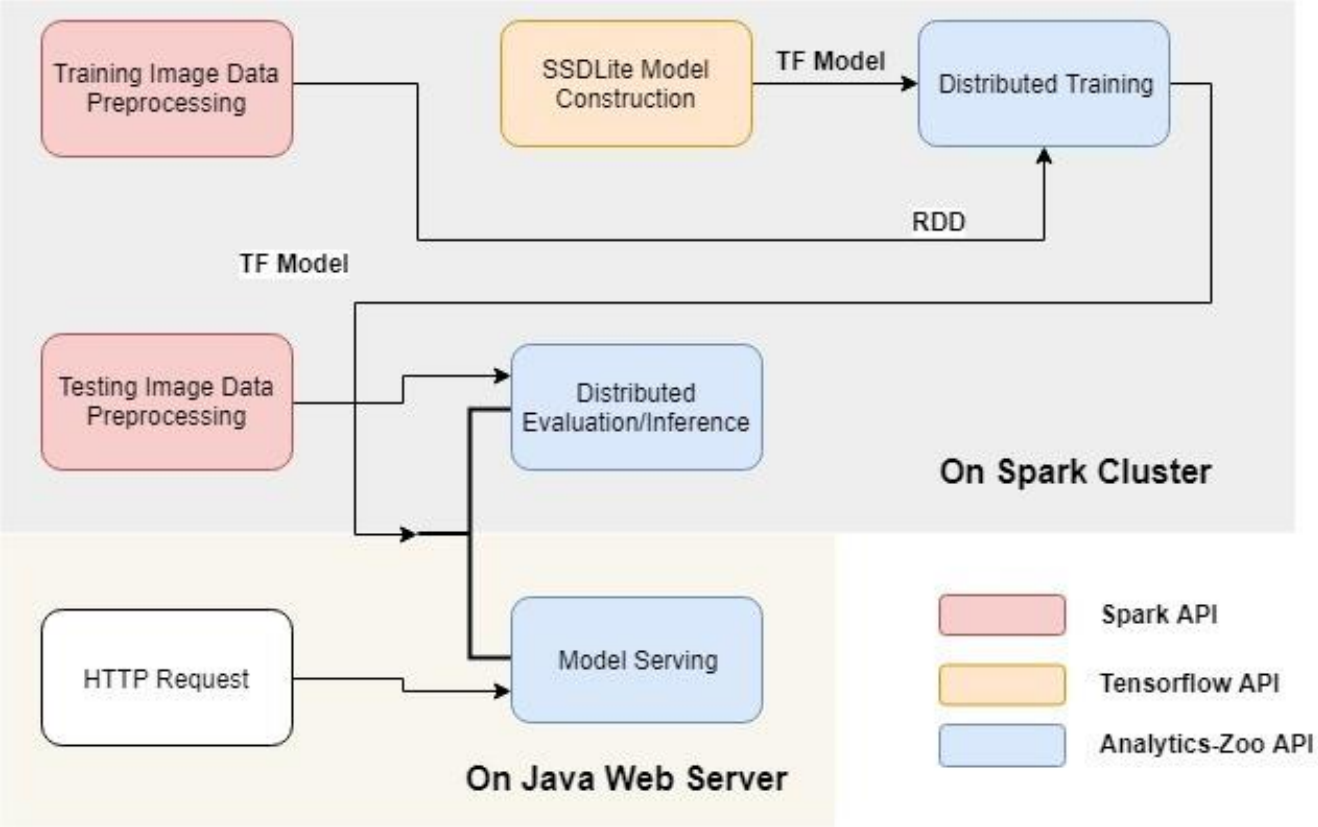

https://software.intel.com/en-us/articles/real-time-product-recommendations-for-office-depot-using-apache-spark-and-analytics-zoo-on
https://conferences.oreilly.com/strata/strata-ca-2019/public/schedule/detail/73079

# 微软AZURE CHATBOT




https://software.intel.com/en-us/articles/use-analytics-zoo-to-inject-ai-into-customer-service-platforms-on-microsoft-azure-part-1
https://www.infoq.com/articles/analytics-zoo-qa-module/

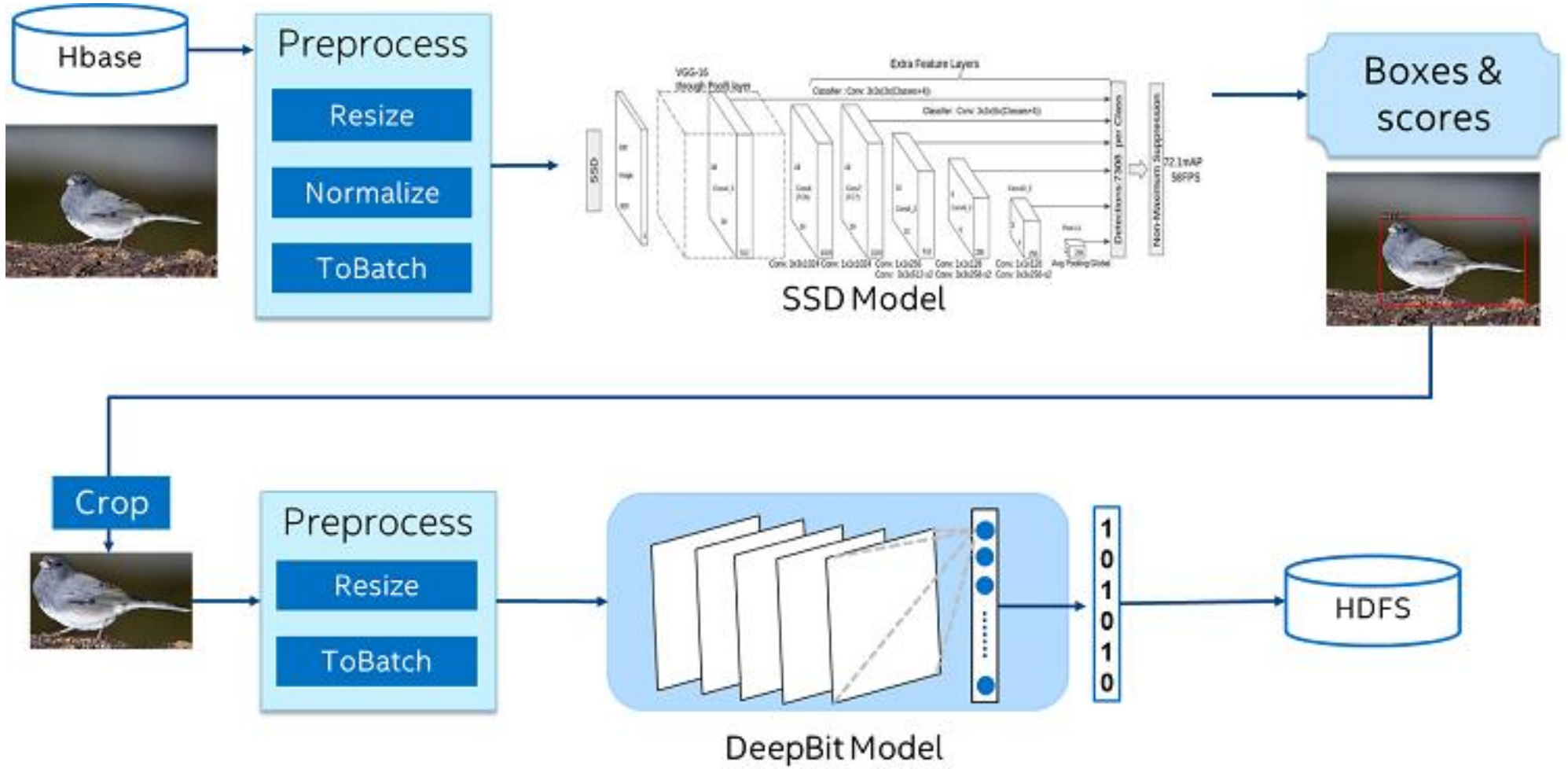
# 美的*：工业视觉检测云平台



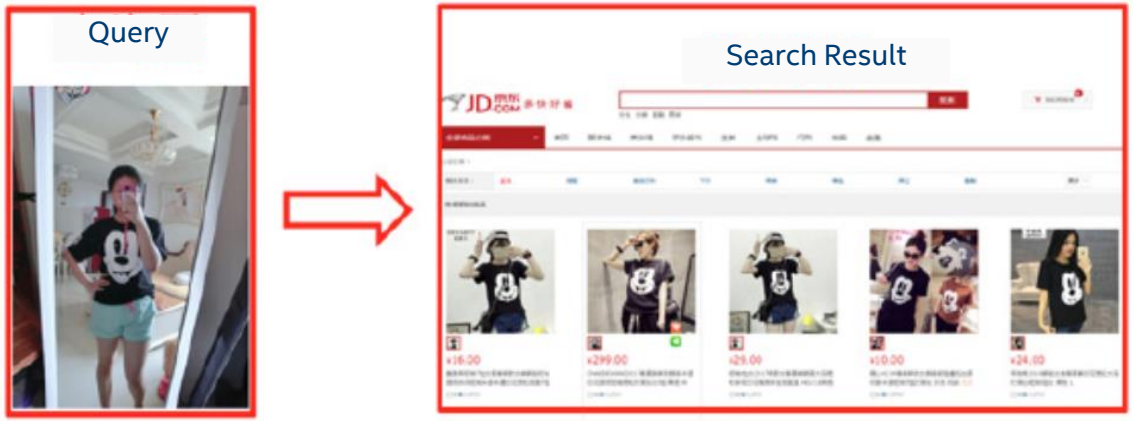

https://software.intel.com/en-us/articles/industrial-inspection-platform-in-midea-and-kuka-using-distributed-tensorflow-on-analytics
https://www.intel.cn/content/www/cn/zh/analytics/artificial-intelligence/midea-case-study.html

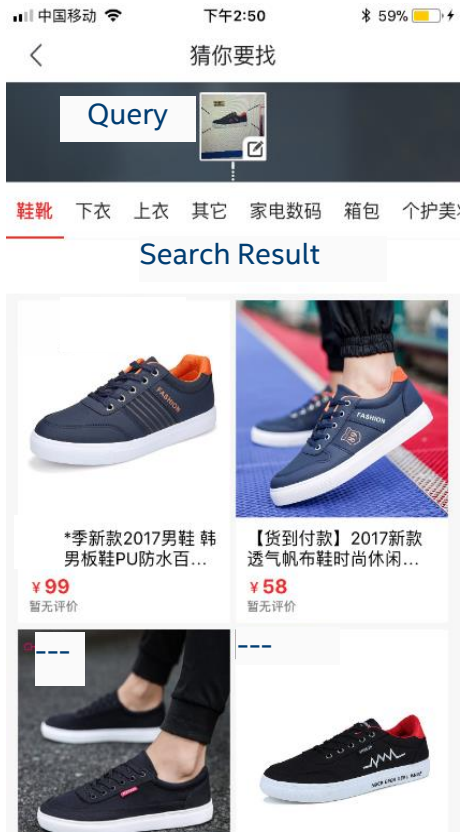
# 美的*：工业视觉检测云平台

# JD.COM： 目标检测和图像特征提取



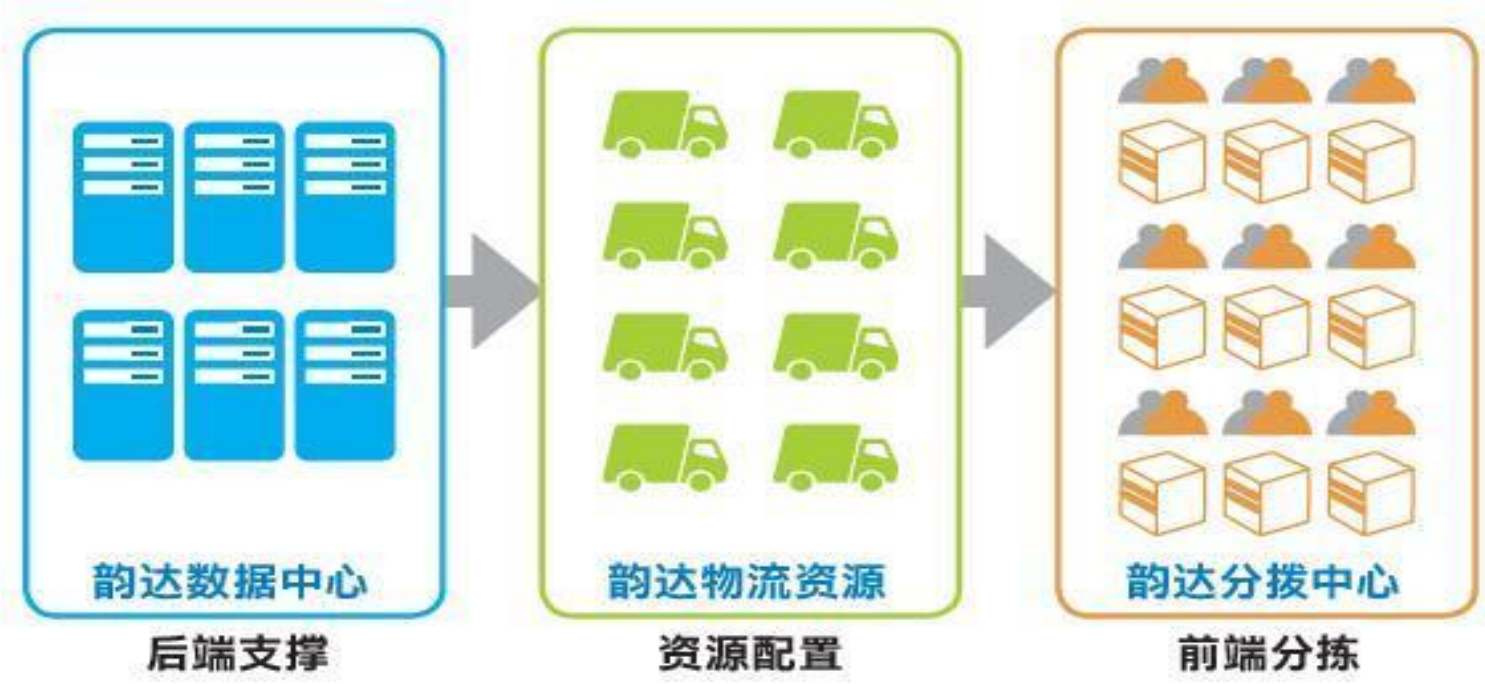
# 相似图片搜索




Source: “Bringing deep learning into big data analytics using BigDL”, Xianyan Jia and Zhenhua Wang, Strata Data Conference Singapore 2017

# 韵达*：基于AI提升快递物流系统运转效率




https://www.intel.cn/content/www/cn/zh/analytics/artificial-intelligence/yunda-brings-quality-change-to-the-express-delivery-industry.html

# 阿里云E-MAPREDUCE SERVICE

## 在阿里云EMR上使用ANALYTICS ZOO

### 集群的创建


https://partners-intl.aliyun.com/help/doc-detail/93155.htm

# And Many More

*Not a full list*

## TECHNOLOGY

bluedata
cloudera
CRAY THE SUPERCOMPUTER COMPANY
databricks
DELL EMC
inspur
GIGASPACES innovate with confidence
Lightbend
Qubole

## CLOUD SERVICE PROVIDERS

Alibaba Cloud aliyun.com
aws
Azure
Tencent 腾讯
Baidu 百度
IBM Cloud
KINGSOFT
Cloud Dataproc

## END USERS

cdhi
Telefonica
中国电信 CHINA TELECOM
THE WORLD BANK
JD.COM 京东
CERN openlab
Midea
韵达 EXPRESS
CISCO
UnionPay 银联

software.intel.com/AIonBigData

# Analytics Zoo 近期将支持的新功能

- 对 Flink 的更多支持
- Cluster Serving 分布式推理 （Part 2 of this session）

PyTorch

- 基于 Apache Spark 的分布式

APACHE Spark

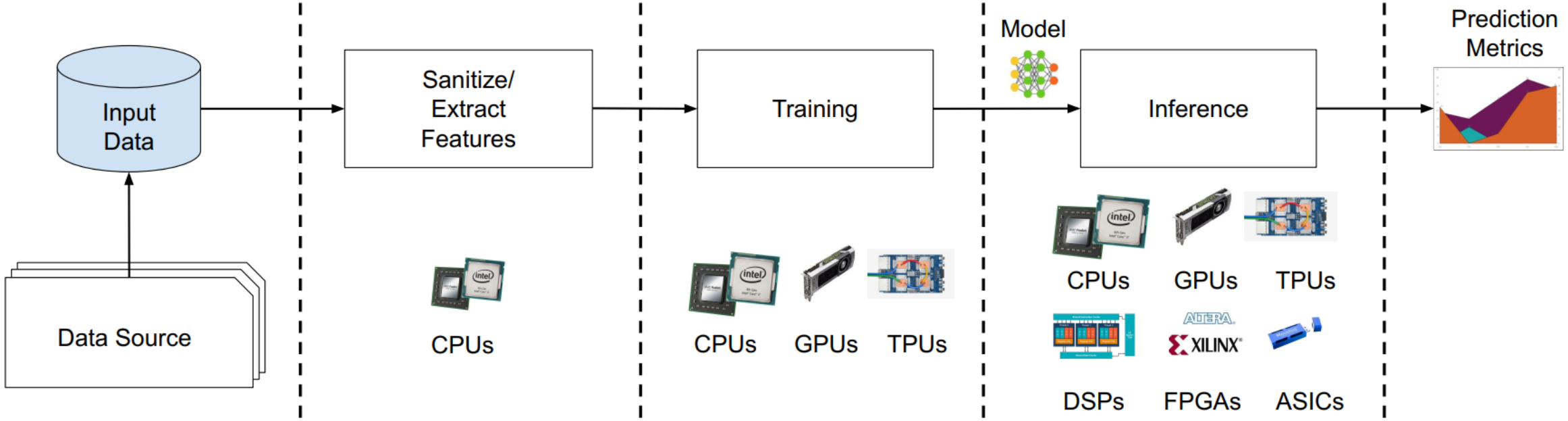
- 直接在 Hadoop 集群上支持 RAY on
  - https://medium.com/riselab/rayonspark-running-emerging-ai-applications-on-big-data-clusters-with-ray-and-analytics-zoo-923e0136ed6a

- AutoML 支持
  - 对时间序列数据（如异常检测或趋势预测）进行自动的特征生成、模型选择和超参调优


Q&A

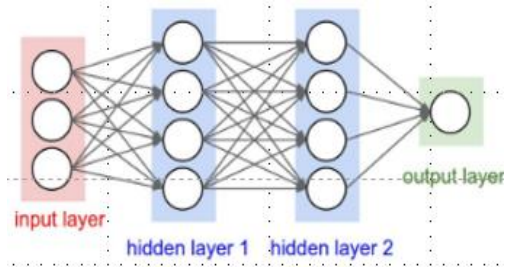

CLUSTER SERVING WITH
ANALYTICS ZOO

# MACHINE/DEEP LEARNING PIPELINE




https://mlperf.org/
VJ Reddi, 2019, MLPerf Inference Benchmark

# MACHINE/DEEP LEARNING PIPELINE




## 训练

Training data → Preprocessing → Training → Postprocessing → model

## 推理

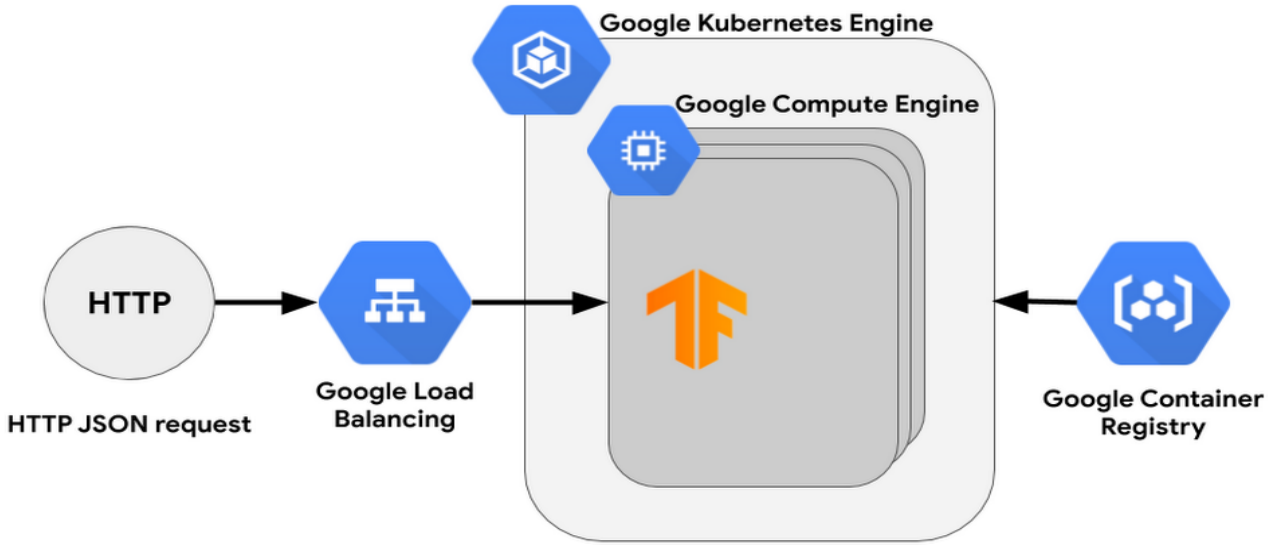
Input Data → Preprocessing → Predict → Postprocessing → Result

End to End (端到端)

# MODEL SERVING

**DL Framework provided Serving**

- TensorFlow Serving
- PyTorch Serving
- MXNet Serving
- PaddlePaddle Serving
- ...

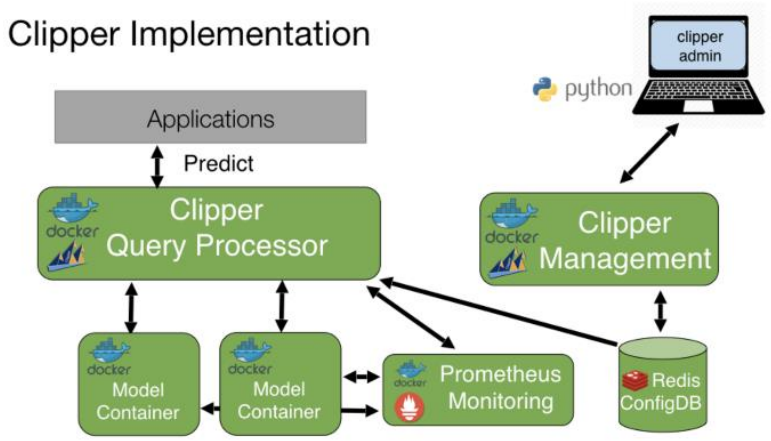



https://cloud.google.com/blog/products/ai-machine-learning/how-to-serve-deep-learning-models-using-tensorflow-2-0-with-cloud-functions

# MODEL SERVING

**General Serving framework**

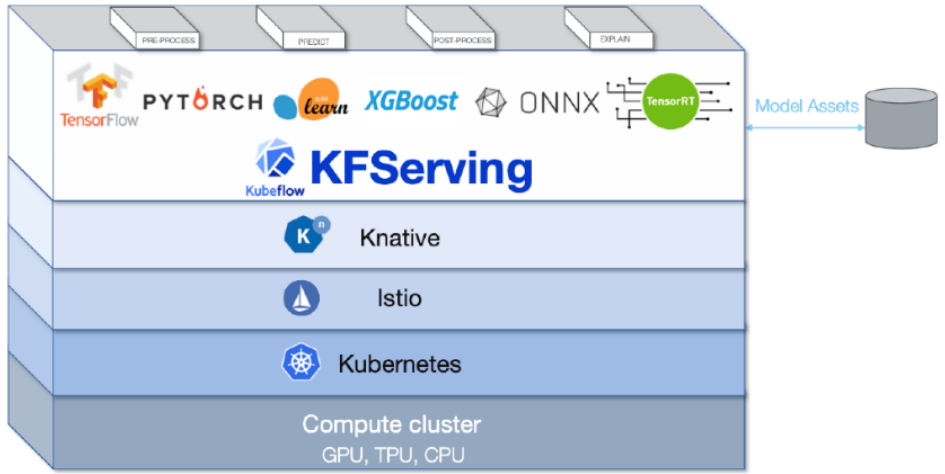
- OpenVINO Serving
- TenorRT Inference Server
- Kubeflow
- MLFlow
- Clipper
- Seldon
- ....




http://clipper.ai/

**Self-implemented Serving Platform**

- High Performance (with CUDA or MKL code)
- But hard to implementation and maintenance

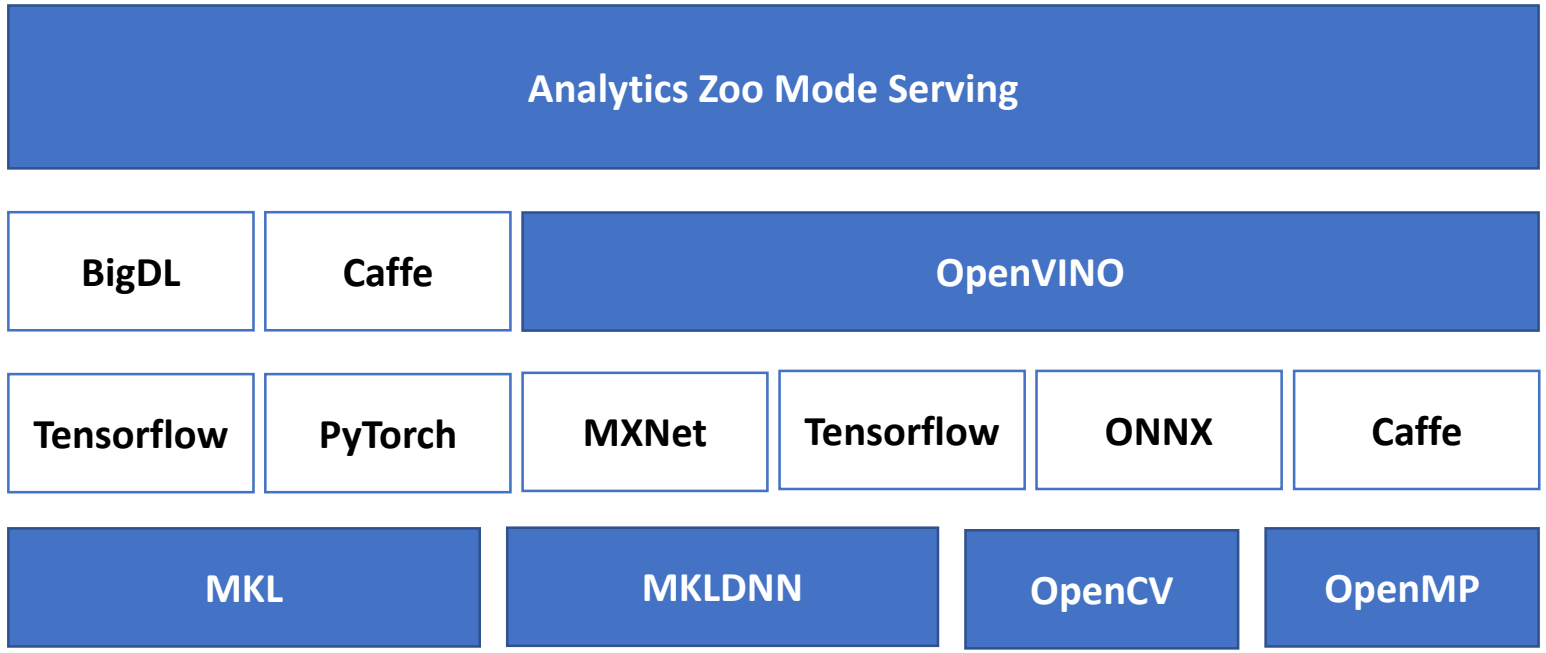



https://github.com/kubeflow/kfserving

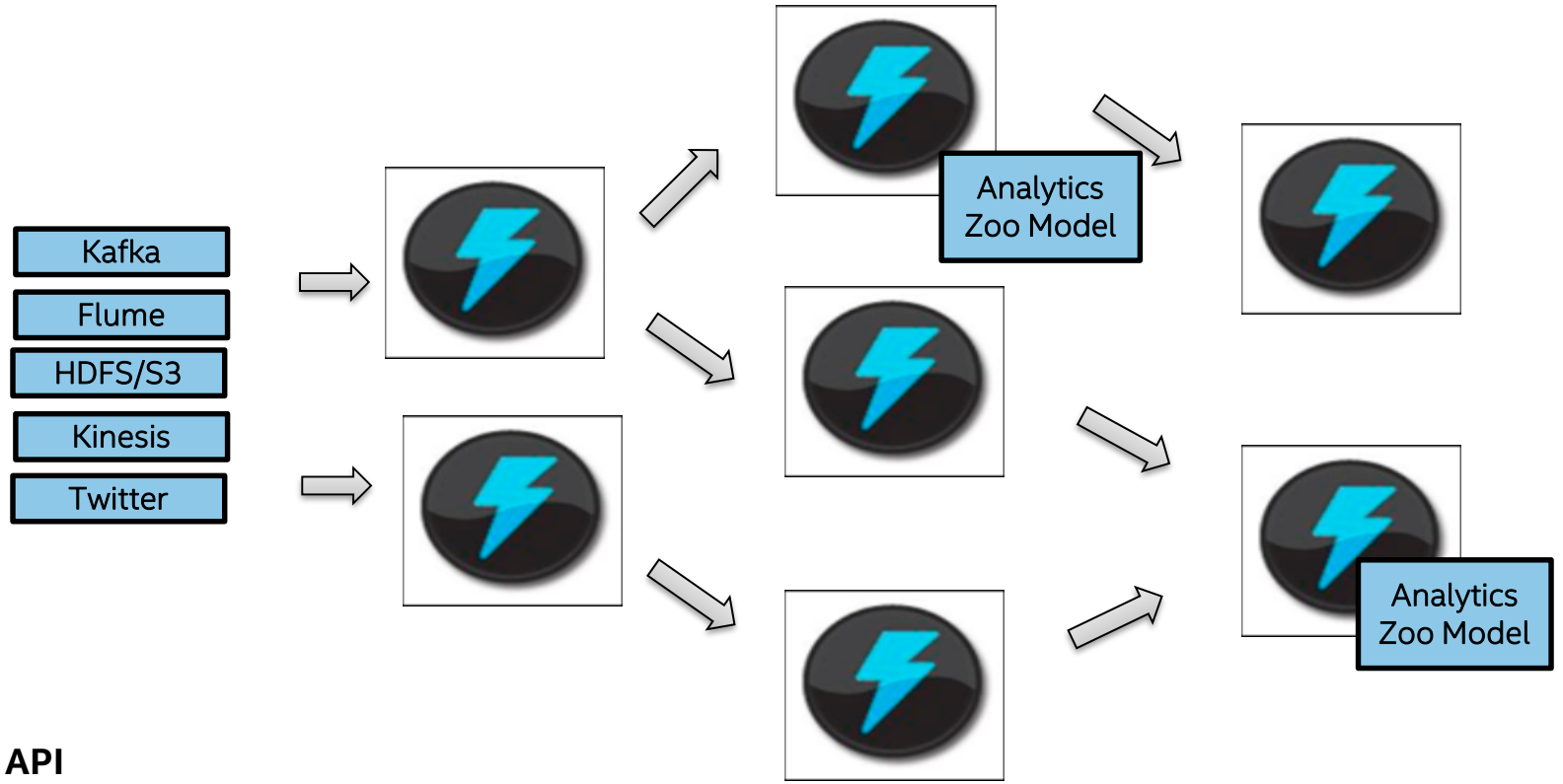
# ANALYTICS-ZOO MODEL SERVING

**支持多种深度学习框架的模型**

- BigDL
- Caffe
- Tensorflow
- PyTorch
- OpenVINO
- Kaldi

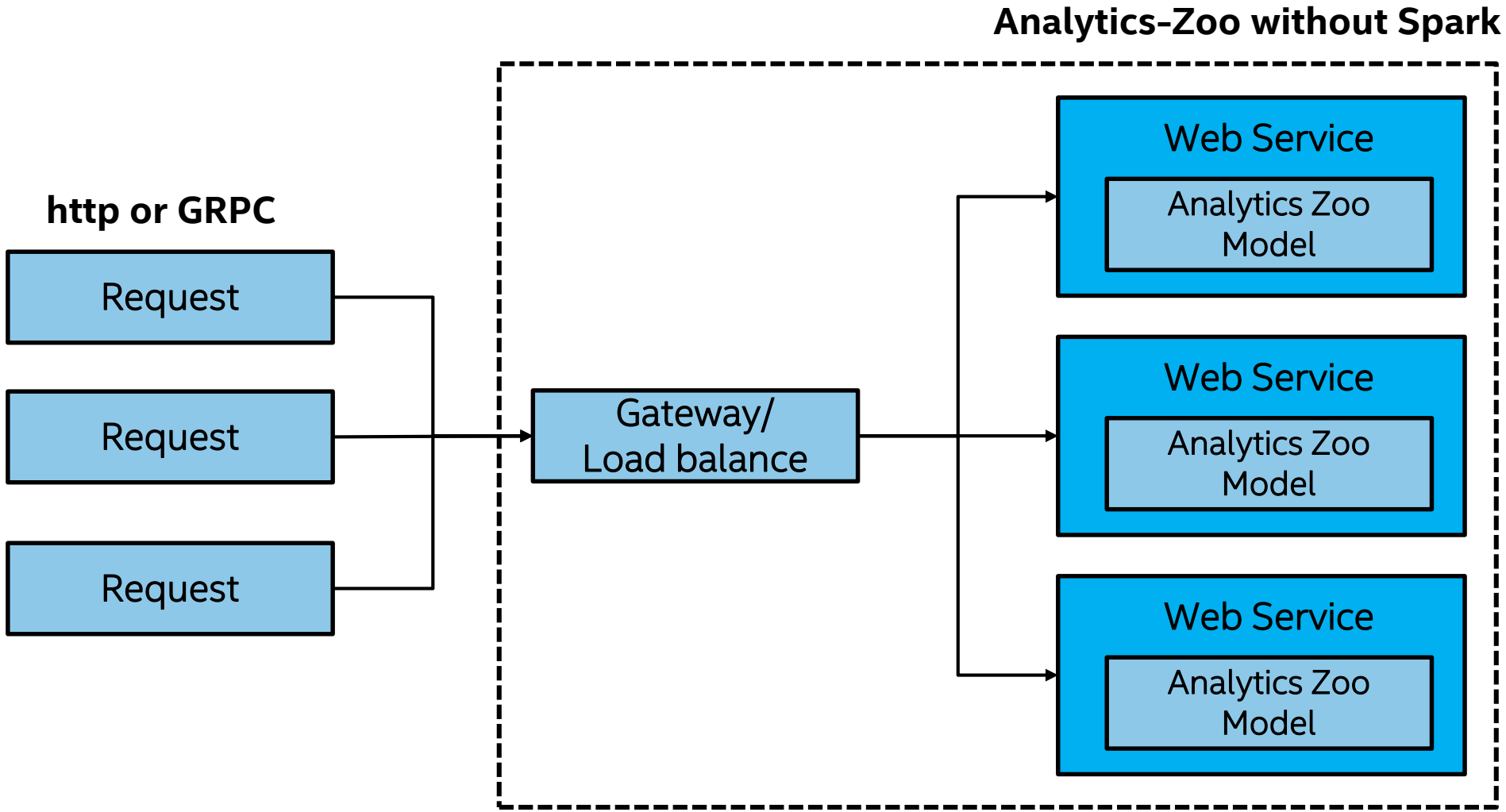


# 分布式、实时(流式)模型推理流水线




- 纯 Java 或 Python API
- 支持 Flink、Kafka、Storm、 Web Service 等
- OpenVINO 和 DL Boost (VNNI) 加速

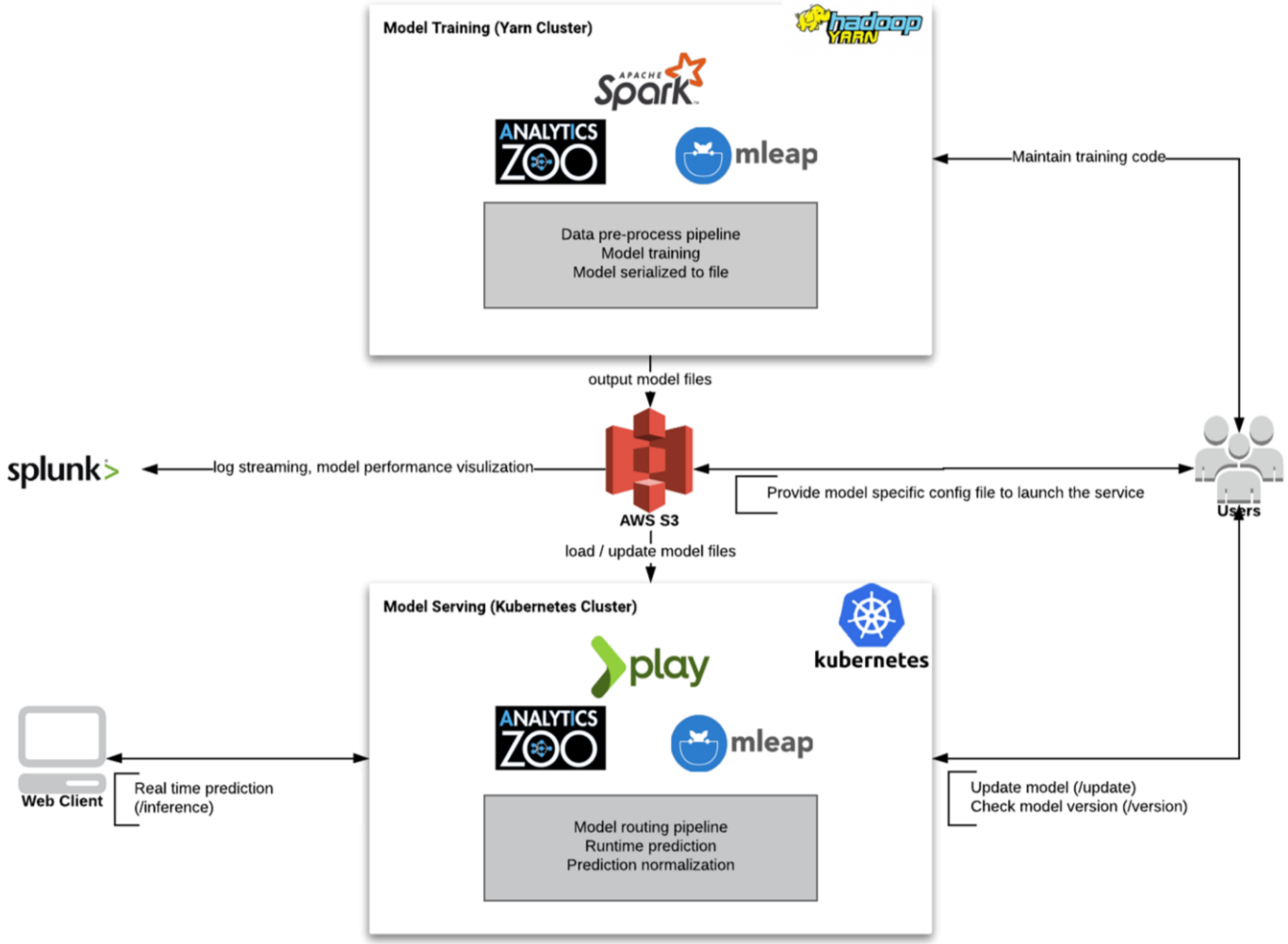
# 基于WEB SERVICE的分布式MODEL SERVING



# 基于WEB SERVICE的分布式MODEL SERVING API

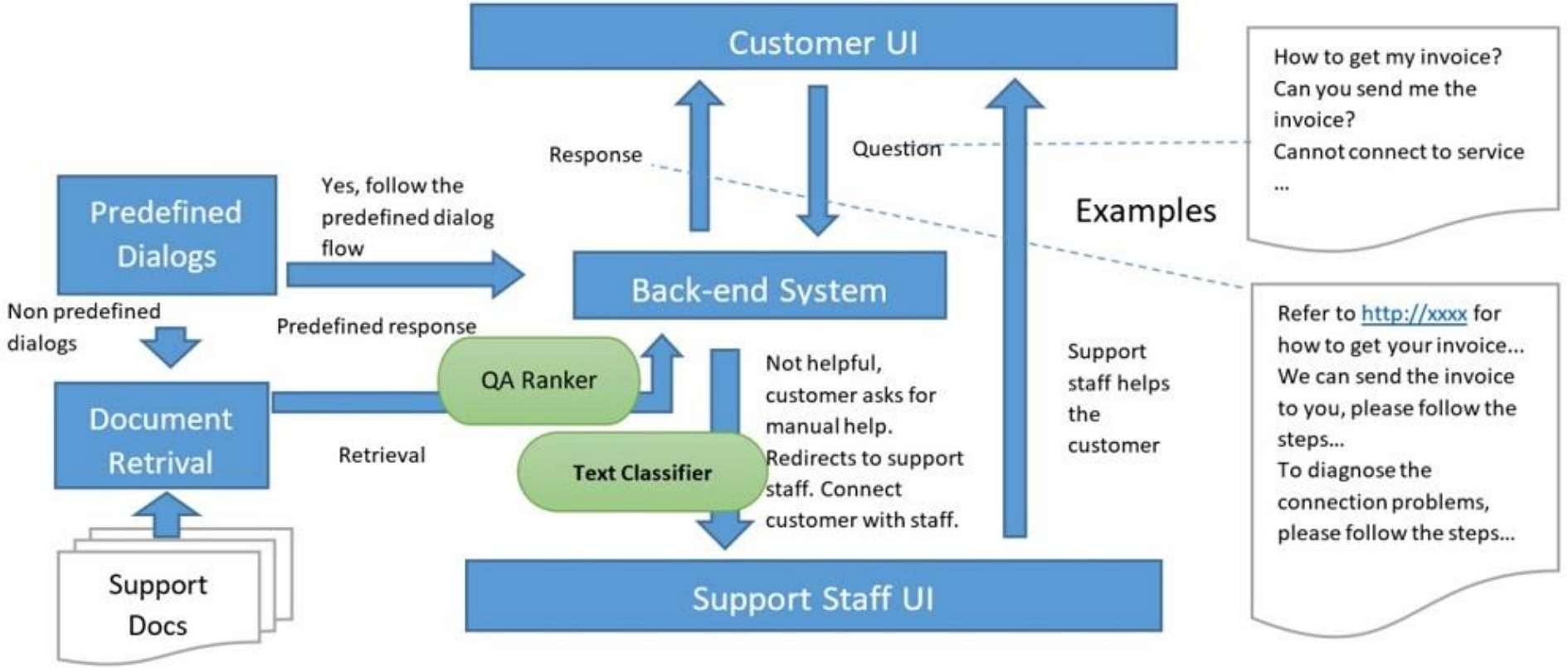
```
import com.intel.analytics.zoo.pipeline.inference.AbstractInferenceModel;

public class TextClassification extends AbstractInferenceModel {
  public RankerInferenceModel(int concurrentNum) {
    super(concurrentNum);
  }
  ...
}

public class ServingExample {
  public static void main(String[] args) throws IOException {
    TextClassification model = new TextClassification();
    model.load(modelPath, weightPath);

    texts = …
    List<JTensor> inputs = preprocess(texts);
    for (JTensor input : inputs) {
      List<Float> result = model.predict(input.getData(), input.getShape());
      ...
  }
}
```

# OFFICE DEPOT*： 基于用户 SESSION 行为的产品推荐



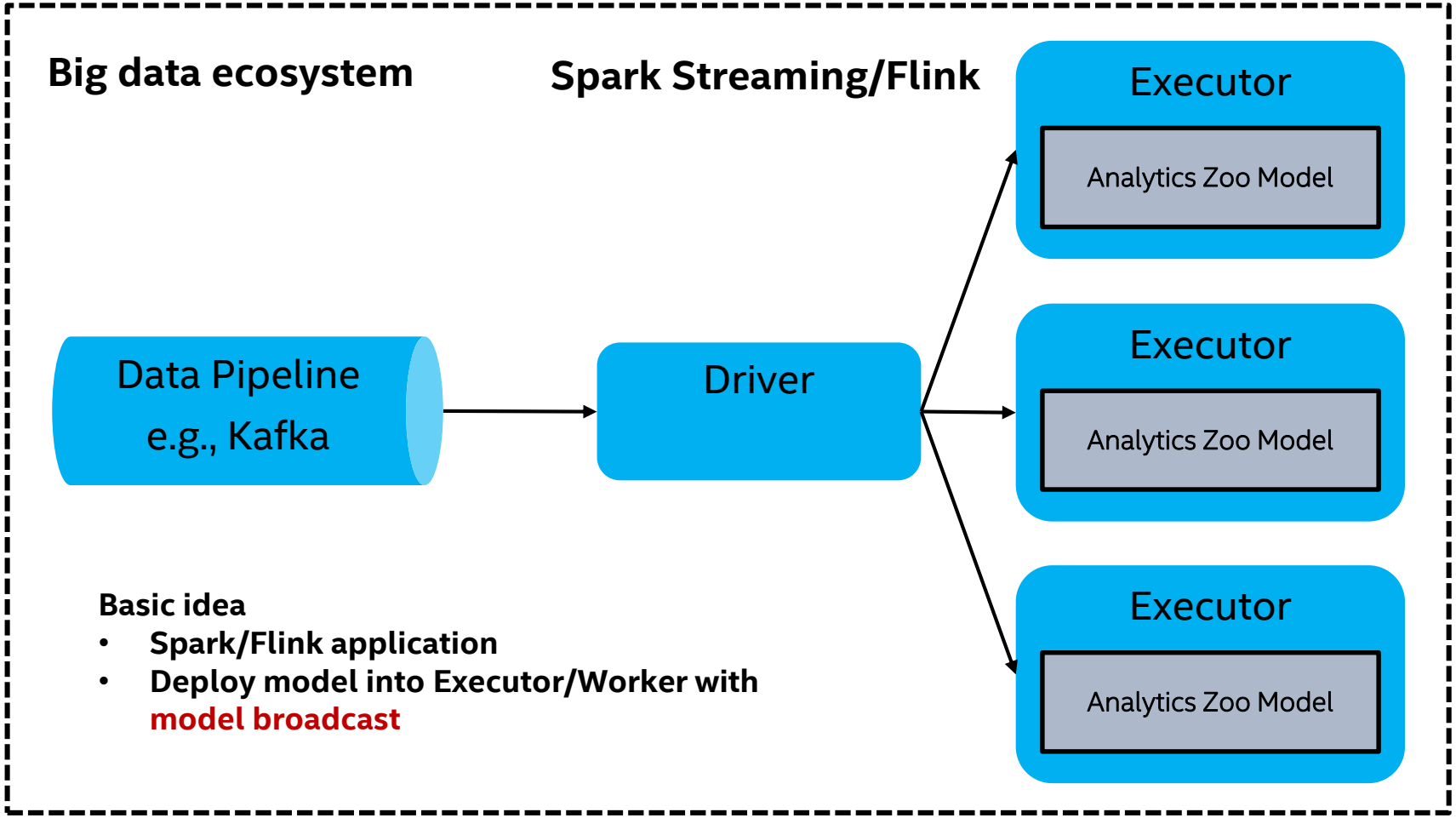

https://software.intel.com/en-us/articles/real-time-product-recommendations-for-office-depot-using-apache-spark-and-analytics-zoo-on
https://conferences.oreilly.com/strata/strata-ca-2019/public/schedule/detail/73079

# 微软AZURE CHATBOT

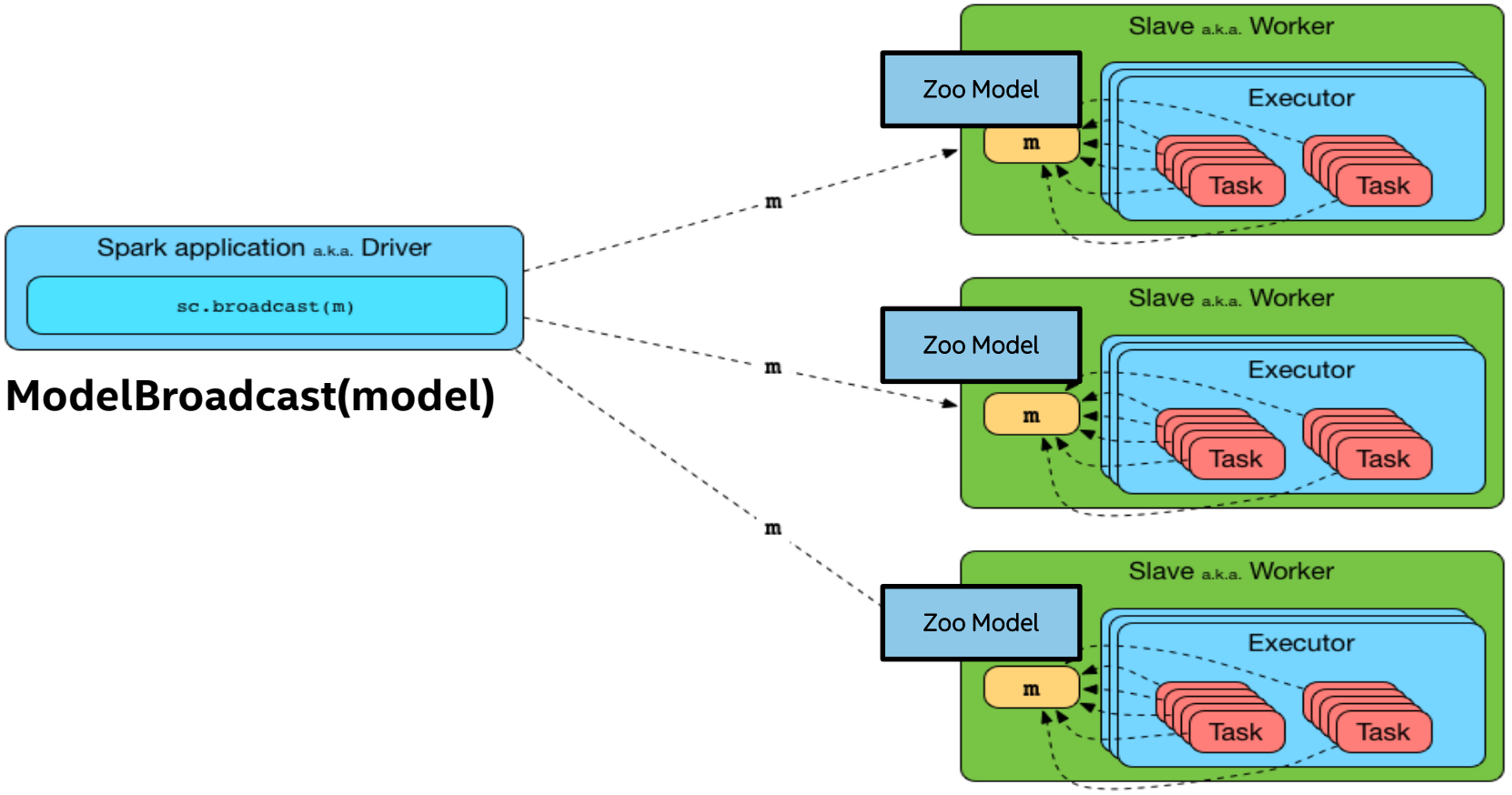



https://software.intel.com/en-us/articles/use-analytics-zoo-to-inject-ai-into-customer-service-platforms-on-microsoft-azure-part-1
https://www.infoq.com/articles/analytics-zoo-qa-module/

# 基于STREAMING的MODEL SERVING



# 基于STREAMING的MODEL SERVING(SPARK)



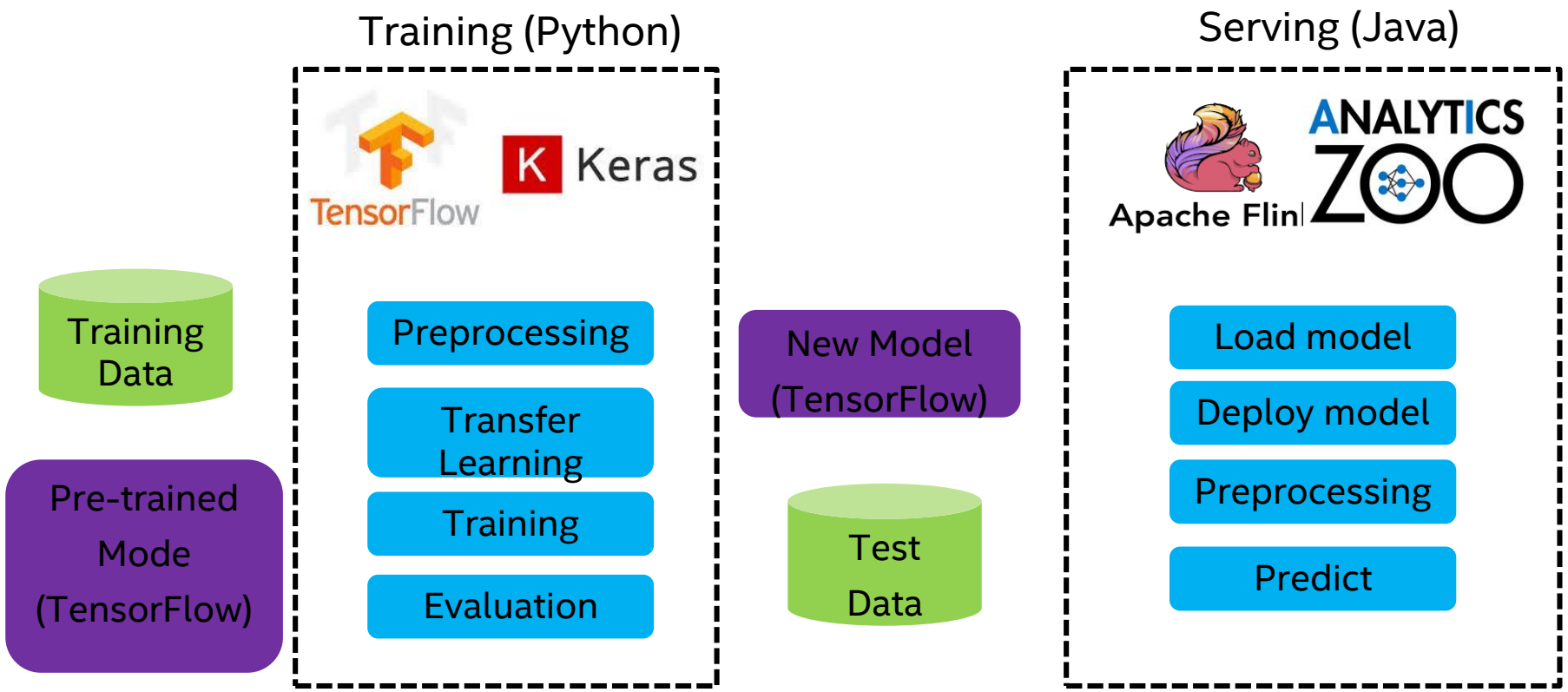
# 基于STREAMING的MODEL SERVING

```
sc = init_nncontext("Streaming Object Detection Example")
ssc = StreamingContext(sc, 3)
# Spark Streaming
lines = ssc.textFileStream(args.streaming_path)

# Analytics-Zoo ObjectDetector API
model = ObjectDetector.load_model(args.model)

def predict(batch_path):
    if batch_path.getNumPartitions() == 0:
        return
    image_set = DistributedImageSet(batch_path.map(read_image_file))
    output = model.predict_image_set(image_set)
    # Save to output
    config = model.get_config()
    visualizer = Visualizer(config.label_map(), encoding="jpg")
    visualizer(output).get_image(to_chw=False)\
        .foreach(lambda x: write_image_file(x, args.output_path))


lines.foreachRDD(predict)
# Start the computation
ssc.start()
# Wait for the computation to terminate
 ssc.awaitTermination()
```

# 阿里APACHE FLINK极客挑战赛


## 这是什么（垃圾）？ 100个类别

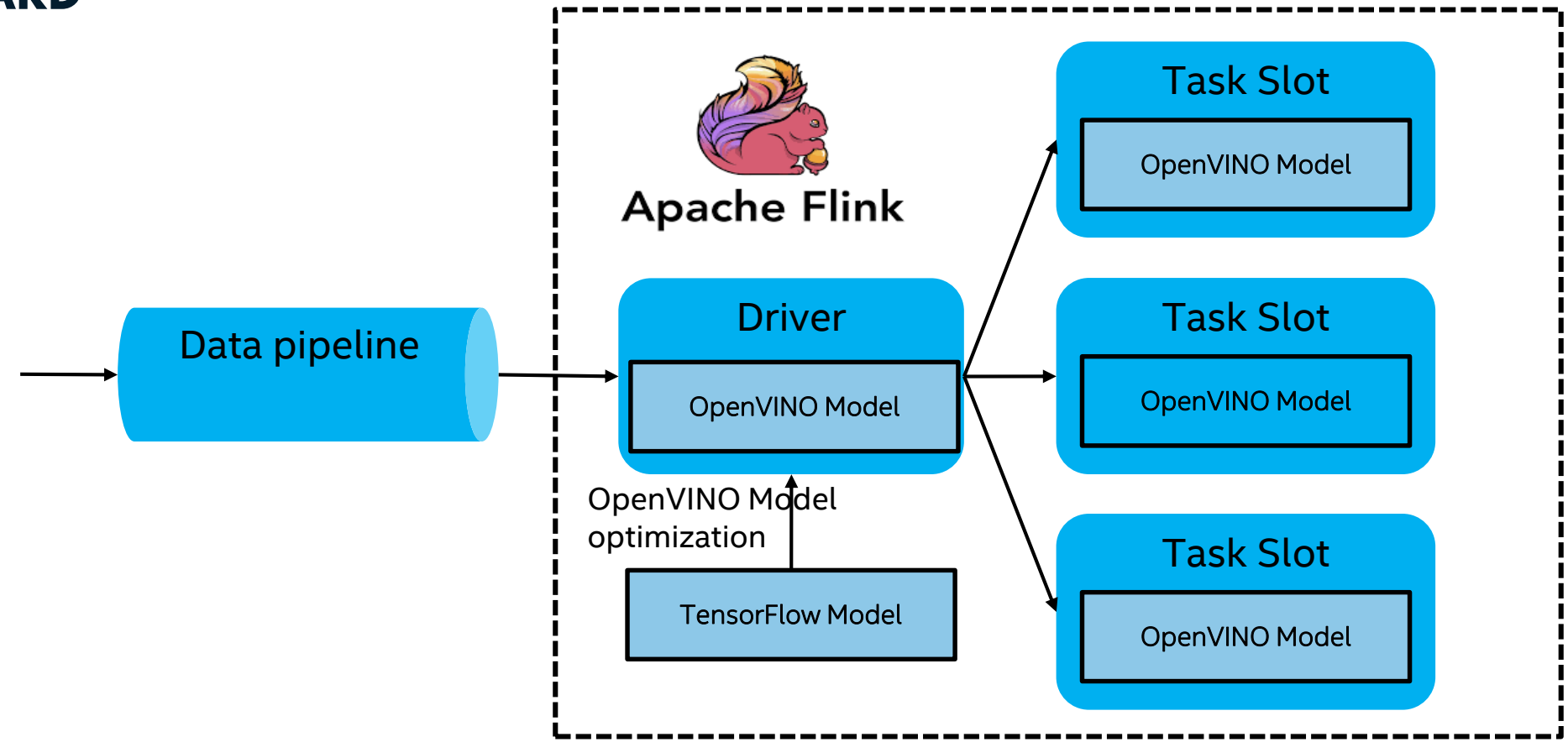
# 阿里APACHE FLINK极客挑战赛



# 基于STREAMING的MODEL SERVING (FLINK)

FLINK FORWARD

在Flink*上使用Analytics Zoo进行大数据分析与深度学习模型推理的架构与实践

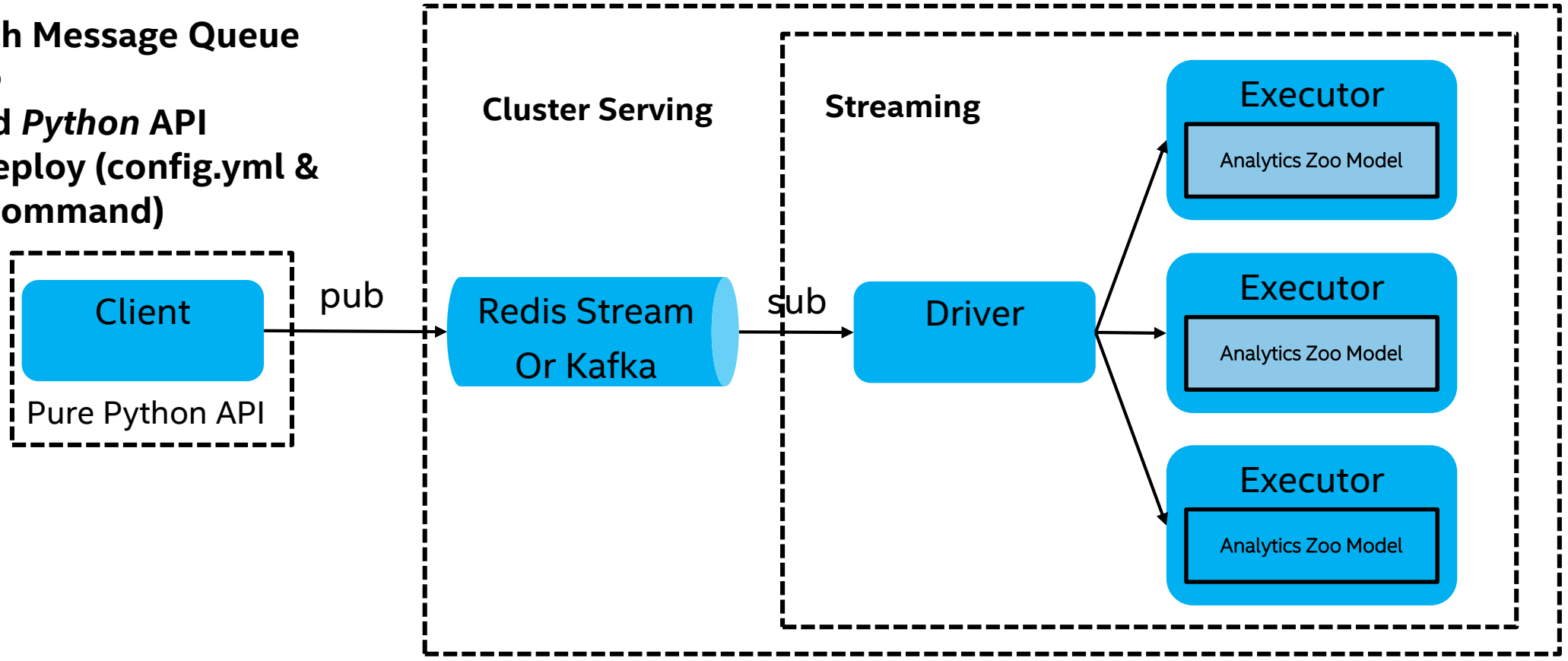


# 能不能更简单一点?

# ANALYTICS-ZOO CLUSTER SERVING

**Pub/Sub with Message Queue**

- ***Real-time***
- **Simplified *Python* API**
- **Easy to deploy (config.yml & one line command)**

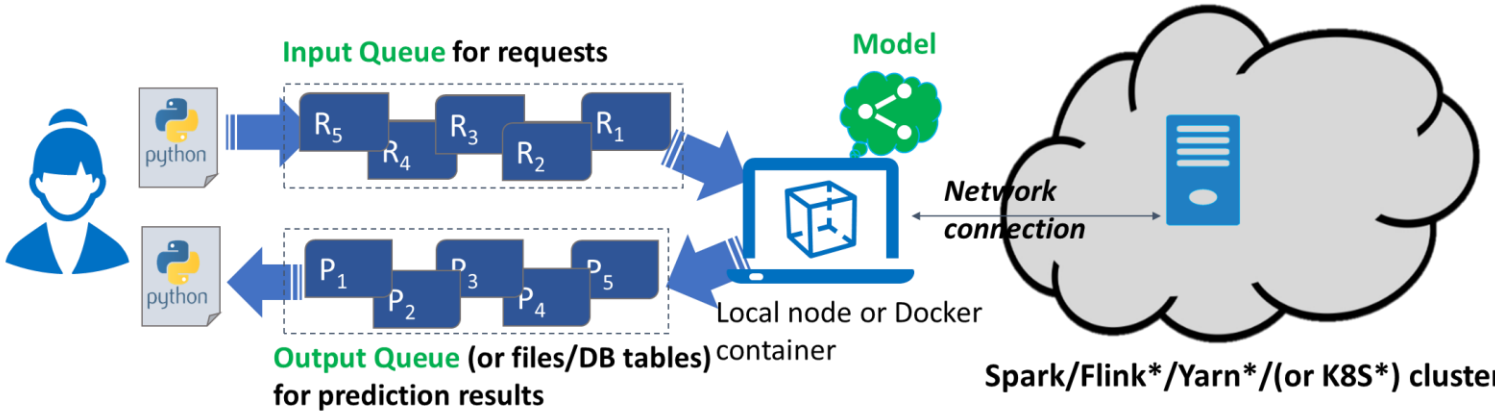


# ANALYTICS-ZOO CLUSTER SERVING

## 部署

√ 一个本地节点或者一个Docker容器

√ 已有的 Yarn/Spark/Flink (or K8s) 集群




## 使用

1 一条命令:

- 启动Docker容器以及*Zoo Cluster Serving*

❑ 此命令指定:

- 输入 和 输出 的队列名字
- 模型 的文件路径
- 预/后处理 的文件路径
- 集群 的访问路径

2 一个简单的Python脚本:

- 将请求数据发送到 Input Queue
- 从 Output Queue (或文件/数据库)获得推理结果

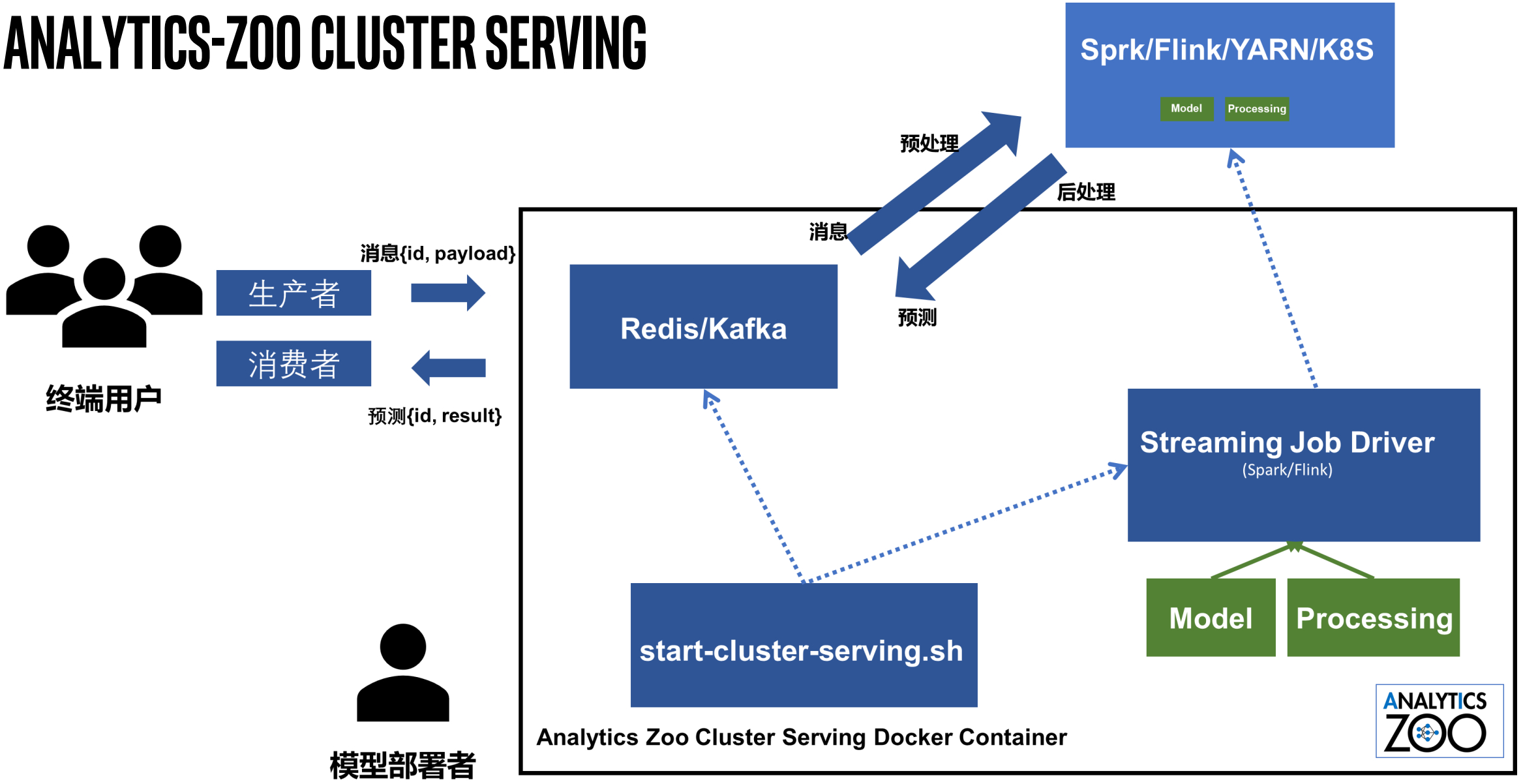
3 Analytics Zoo 在集群上自动执行分布式、实时 （流式）模型推理

- 支持 TensorFlow*, Keras*, PyTorch*, Caffe*, BigDL 和 OpenVINO 的模型, 可使用 Int8 加速
- 通过Spark/Flink* 线性扩展

√ 可扩展的分布式推理由Analytics Zoo托管

√ 用户无需为开发和部署复杂的分布式推理方案而费心

# ANALYTICS-ZOO CLUSTER SERVING



# ANALYTICS-ZOO CLUSTER SERVING

## Pure Python Client

```
from zoo.serving.client.utils.helpers import RedisQueue
import cv2

if __name__ == "__main__":
    # if you do not specify config_path, it will use default config
    config_path = "/path/to/analytics-zoo-cluster-serving/config.yaml"
    redis_queue = RedisQueue(config_path)

    # following lines demonstrate how to push data to redis
    img_path = "/path/to/image"
    # pre-processing in client is optional
    img = cv2.imread(img_path)
    img = cv2.resize(img, (224, 224))
    redis_queue.enqueue_image(img)

    # following lines demonstrate how to get data from redis
    redis_queue.get_results("result:*")
```

# ANALYTICS-ZOO CLUSTER SERVING

## YML Configuration

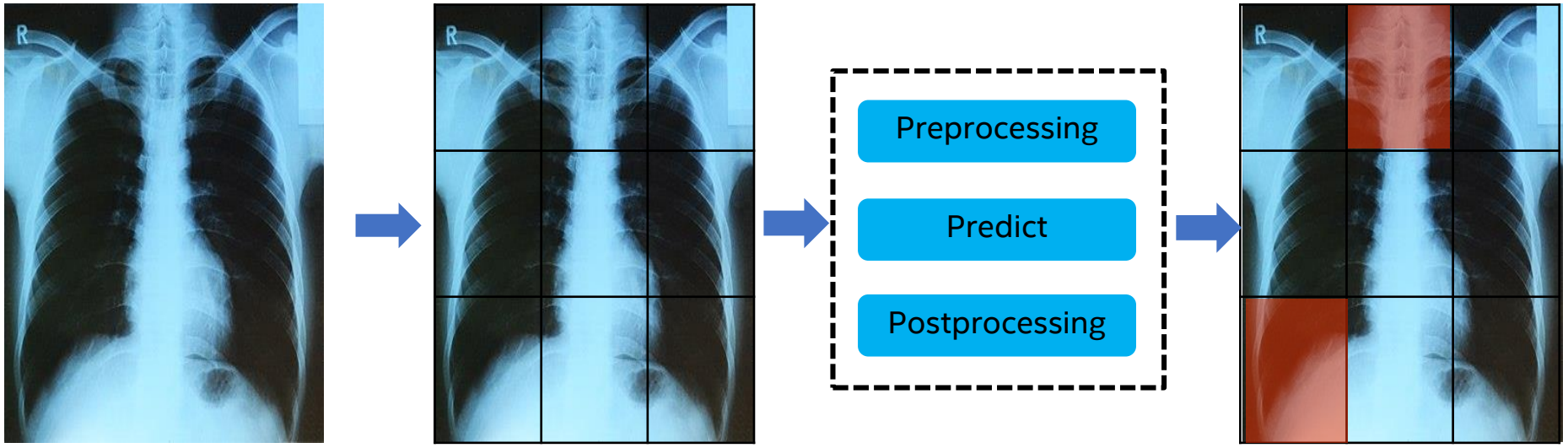
```
## Analytics-zoo Cluster Serving
model:
  # model path must be set
  path: resources
data:
  # redis address
  src: XXXXXX:6379
  shape: 3, 224, 224

  # default, image-classification
  task:

spark:
  # default, local[*]
  master: spark://XXXX:7077

params:
  # default, 4
  batch_size: 8

  # default, mkldnn
  engine_type
```

# ANALYTICS-ZOO CLUSTER SERVING

## User Customized Serving Pipeline

```
from zoo.pipeline.inference import InferenceModel
from zoo.common.nncontext import init_nncontext
from zoo.feature.image import *
from zoo.pipeline.nnframes import *

sc = init_nncontext("OpenVINO Python resnet_v1_50 Inference Example")
# pre-processing
infer_transformer = ChainedPreprocessing([ImageBytesToMat(),
                                          ImageResize(256, 256),
                                          ImageCenterCrop(224, 224),
                                          ImageMatToTensor(format="NHWC", to_RGB=True)])
images = ImageSet.read(img_path, sc).\
                transform(infer_transformer).get_image().collect()
# load openvino model
model = InferenceModel()
model.load_openvino(model_path, weight_path)

# predict
for batch in images:
   model.predict(batch)
```

# CLUSTER SERVING加速医疗影像分析



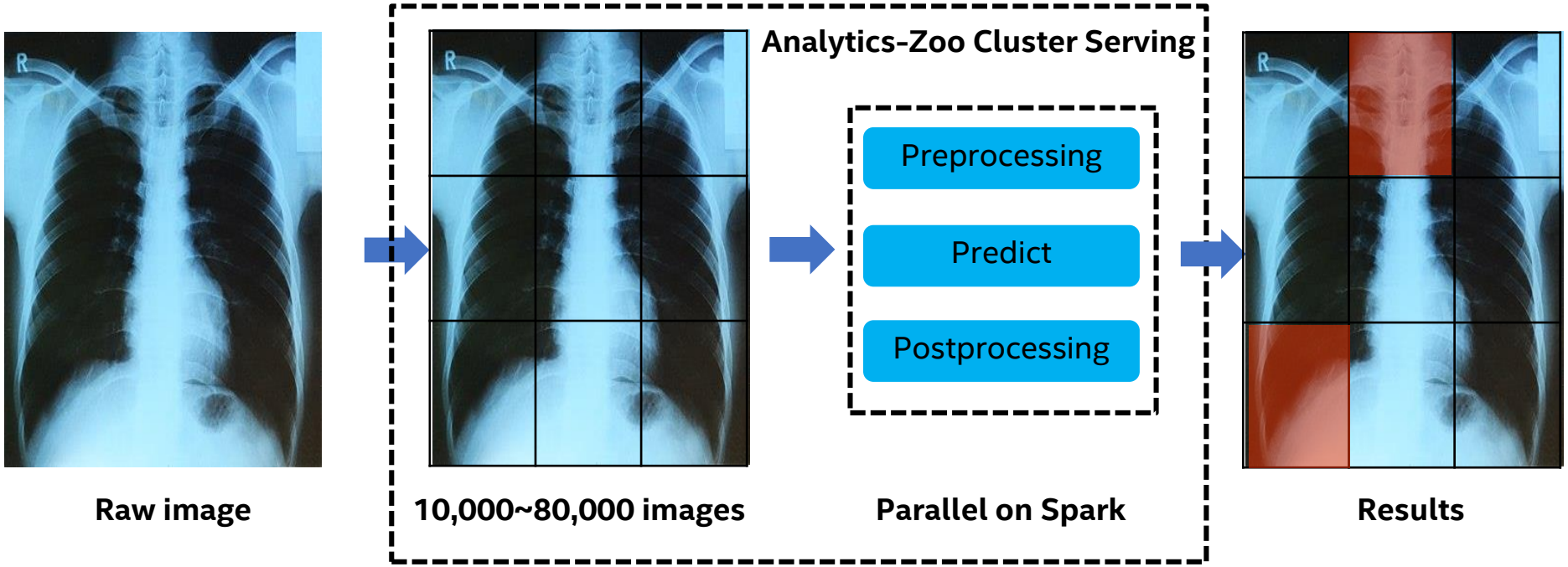

Raw image | 10,000~80,000 images | Python Code & API | Results

**Result is good, but performance is unacceptable (1~2 hours for each raw image), and its really hard to scale**

**Bottleneck: preprocessing (split, crop, resize and normalization， 90~99%)**

https://en.wikipedia.org/wiki/X-ray

# CLUSTER SERVING 加速医疗影像分析




**(1~2 hours ➔ 5~20 seconds) with Analytics-Zoo Cluster Serving**

- **Parallel image preprocessing with Spark & Analytics-Zoo (OpenCV)**
- **Parallel inference/predict with Analytics-Zoo (Caffe & MKLDNN)**
- **Easy to implementation and easy to scale**
- **FP32 model, can achieve better performance with int8**

https://en.wikipedia.org/wiki/X-ray

# End-to-End Big Data and AI Pipelines

Seamless Scaling from Laptop to Production


## Unified Analytics + AI Platform

Distributed TensorFlow*, Keras*, PyTorch* & BigDL on Apache Spark*

https://github.com/intel-analytics/analytics-zoo


intel®
AI

# LEGAL NOTICES AND DISCLAIMERS

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.