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

史栋杰

英特尔资深软件架构师

FLINK FORWARD # ASIA
实时即未来 # Real-time Is The Future

FLINK FORWARD

自我介绍

Self Introduction

史栋杰，英特尔资深软件架构师。多年从事企业级计算、风控、大数据分析、云计算容器编排、数据分析与人工智能领域的研发，英特尔开源框架 BigDL 与 Analytics Zoo 的贡献者之一。

Contents
目录

大规模人工智能应用面临的挑战

AI production at scale is facing lots of challenges

01

以数据为中心的世界

The Data-Centric World

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

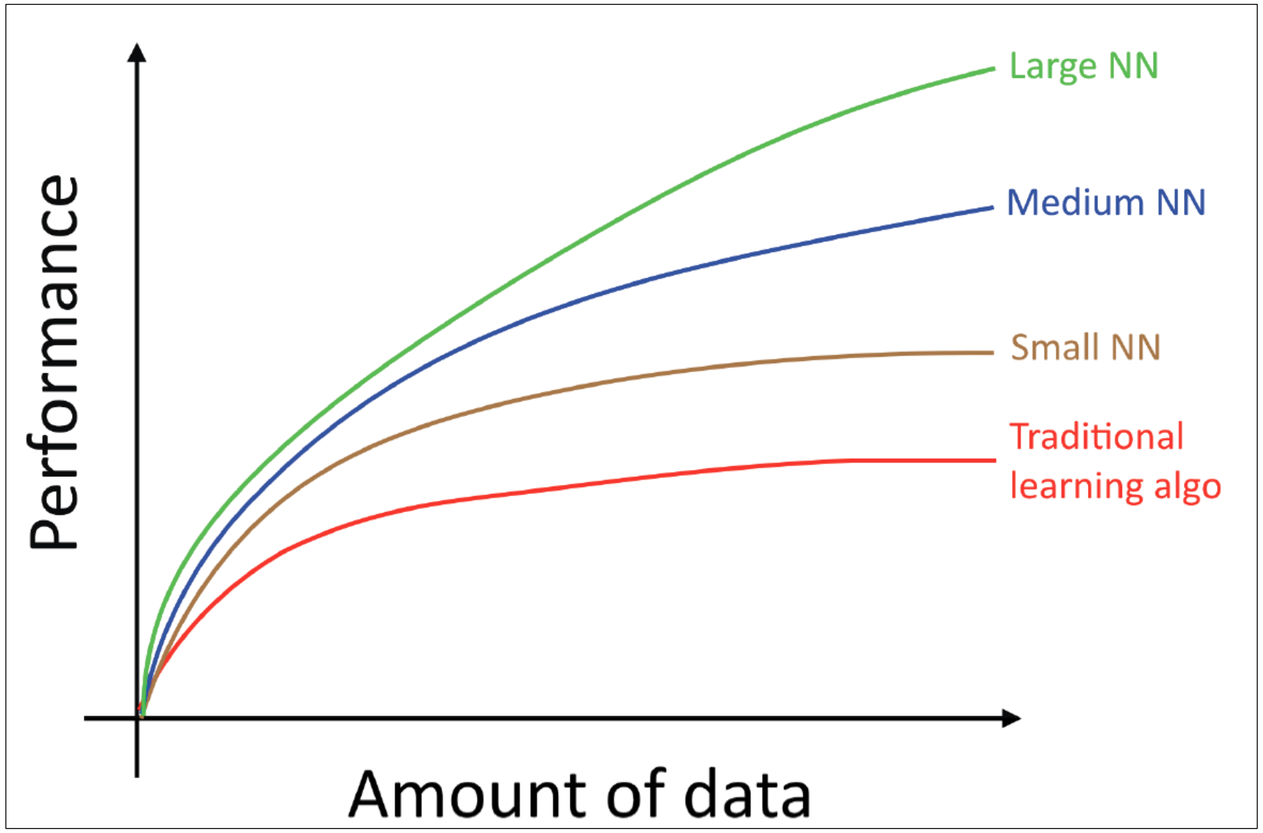
经过了分析

大规模人工智能应用

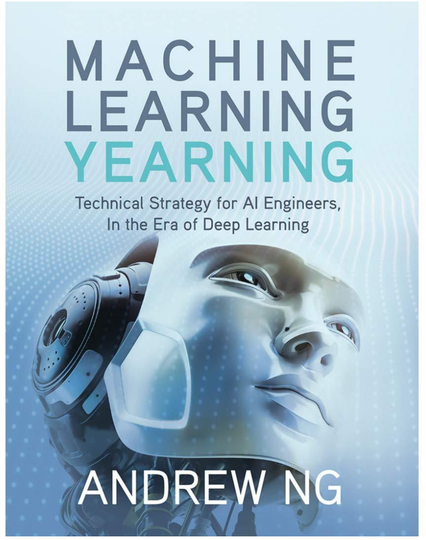
AI Production at Scale

数据驱动深度学习和人工智能应用

Data drives deep learning and AI production



“Machine Learning Yearning”, Andrew Ng, 2016

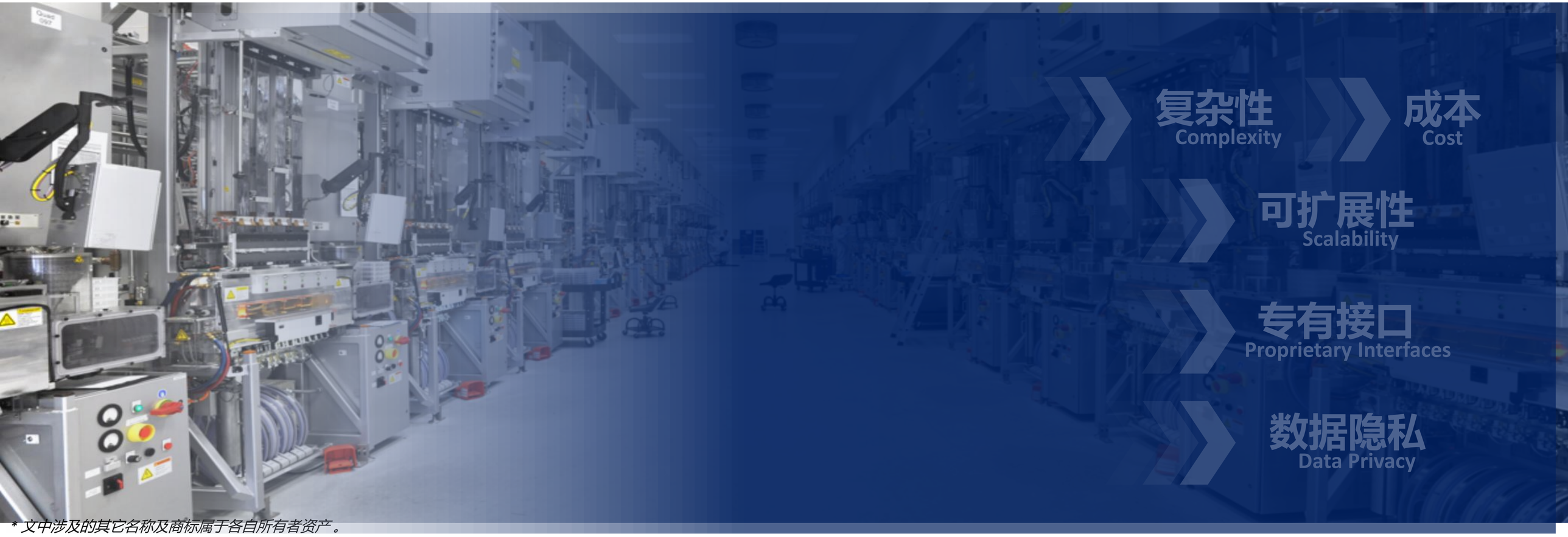


大规模人工智能应用

AI Production at Scale

正面临巨大的挑战

Facing Lots of Challenges



大规模人工智能应用

AI Production at Scale

正面临巨大的挑战

Facing Lots of Challenges

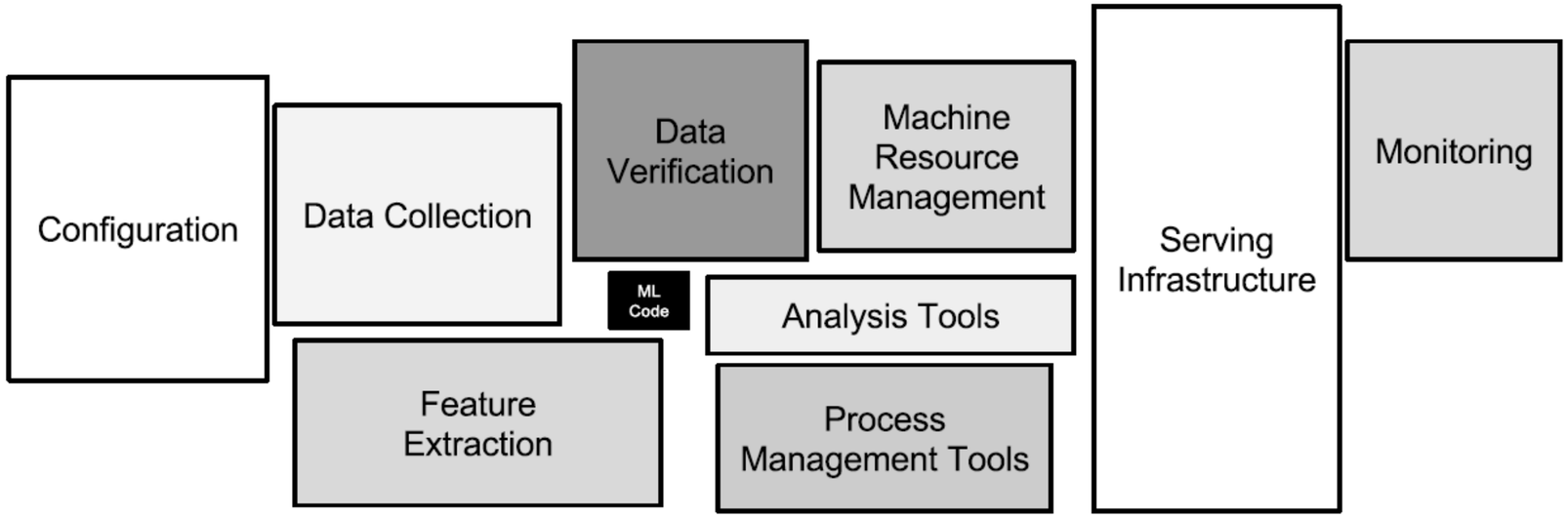


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

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

统一的大数据分析及人工智能

Integrated Data Analytics and AI

02

统一的大数据分析及人工智能

Integrated Data Analytics and AI

获取 / 存储 Source/Store → **清洗 / 准备** Clean/Prepare → **分析 / 建模** Analyze /Modeling → **部署 / 可视化** Deploy /Visualize

集成的数据流水线

Unified Data Pipeline

Apache Flink

hadoop HDFS

APACHE Spark

Apache Flink

Spark MLlib

APACHE HBASE

APACHE kafka
A distributed streaming platform

APACHE KUDU

RAY

TensorFlow

BigDL

ANALYTICS ZOO

数据管理 Data Management

数据分析 Data Analysis

数据科学及人工智能 Data Science and AI

FLINK FORWARD

大数据上的人工智能

AI on Big Data

BigDL

高性能深度学习框架

High-Performance Deep Learning Framework for Apache Spark*

software.intel.com/bigdl

ANALYTICS ZOO

统一的分析 + 人工智能平台

Integrated Analytics + AI Toolkit

分布式

TensorFlow、PyTorch、Keras 和 BigDL

高级流水线、参考用例、人工智能模型、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

Accelerating DATA Analytics + AI Solutions DEPLOYMENT At SCALE

** 文中涉及的其它名称及商标属于各自所有者资产。*

统一的数据分析和AI流水线

End-to-End Big Data Analytics and AI Pipeline

端到端、从原型到生产化部署的无缝扩展

Seamless Scaling from Laptop to Production

在笔记本电脑上使用样本数据构建原型
Prototype on laptop using sample data

在集群上使用历史数据运行模型试验
Experiment on clusters with history data

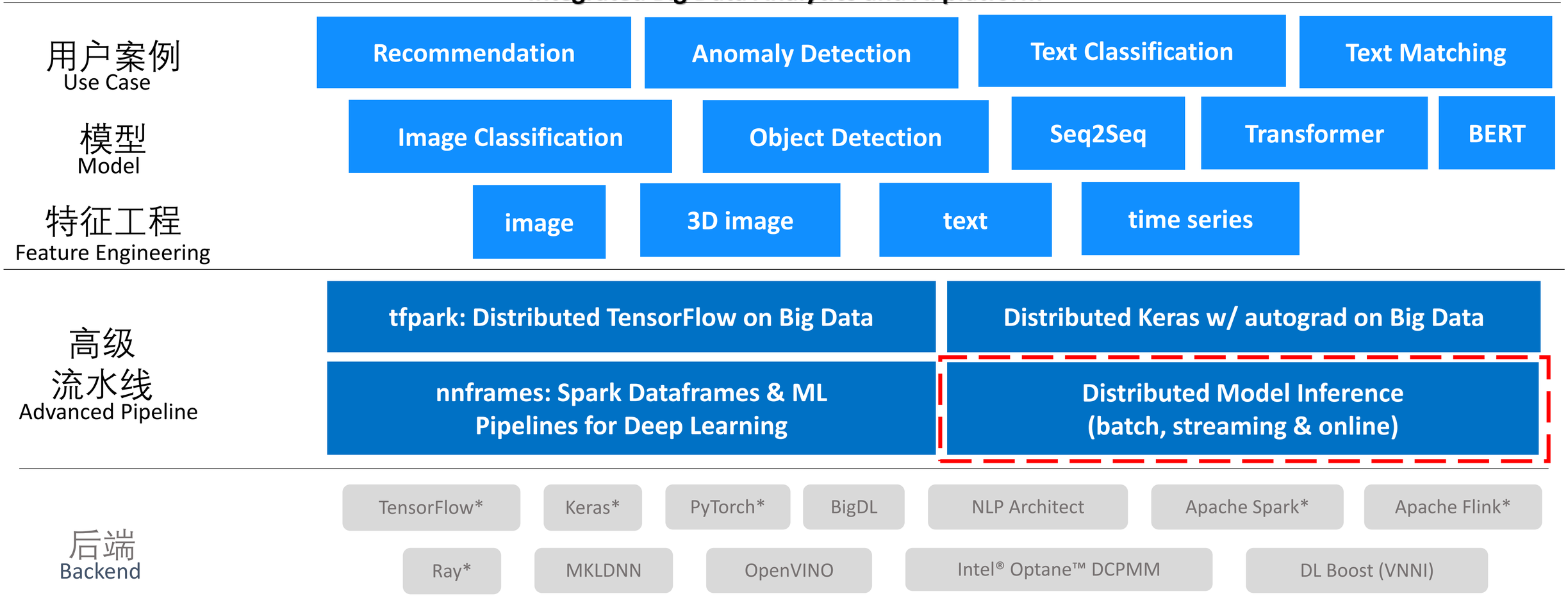
在分布式生产环境中部署
Production deployment w/ distributed data pipeline

- 从笔记本电脑到分布式集群几乎无需任何代码更改 **“Zero” code change from laptop to distributed cluster**
- 无需数据拷贝，直接访问生产大数据系统 **Directly access production data without data copy**
- 高效构建端到端的数据分析+ AI 流水线原型 **Easily prototype the end-to-end pipeline**
- 无缝扩展部署到大数据集群及生产环境 **Seamlessly deployed on production big data clusters**

Analytics Zoo

统一的大数据分析+人工智能平台

Integrated Big Data Analytics and AI platform

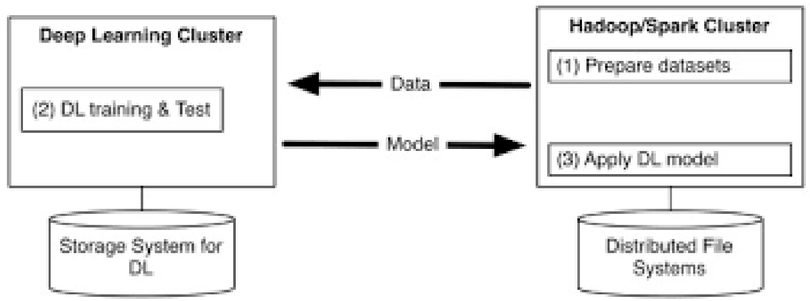


https://github.com/intel-analytics/analytics-zoo

分布式 TensorFlow* 流水线

Distributed TensorFlow* Pipeline

- Data loading, processing and feature engineering with Big Data
- Deep learning model development using TensorFlow* or Keras*
- Distributed training / inference on Big Data




```
#load data
train_data = hadoopFile(…).map(…)
dataset = TFDataset.from_rdd(train_rdd,…)
```

用大数据计算框架载入数据以及预处理数据或特征工程

```
#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))
```

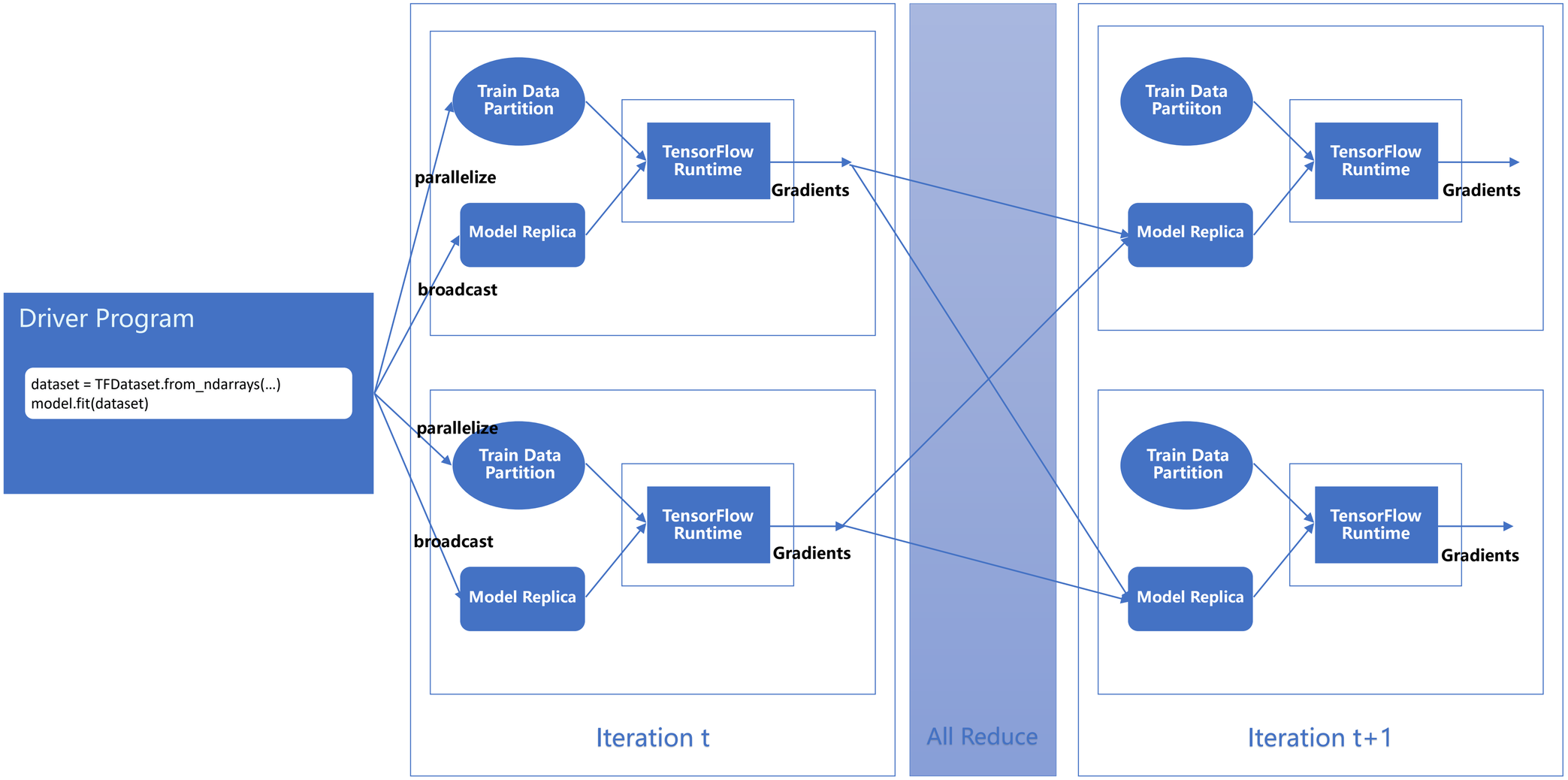
用 TensorFlow*或Keras*定义深度学习模型

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

在大数据上分布式训练或者推理

分布式 TensorFlow* 流水线

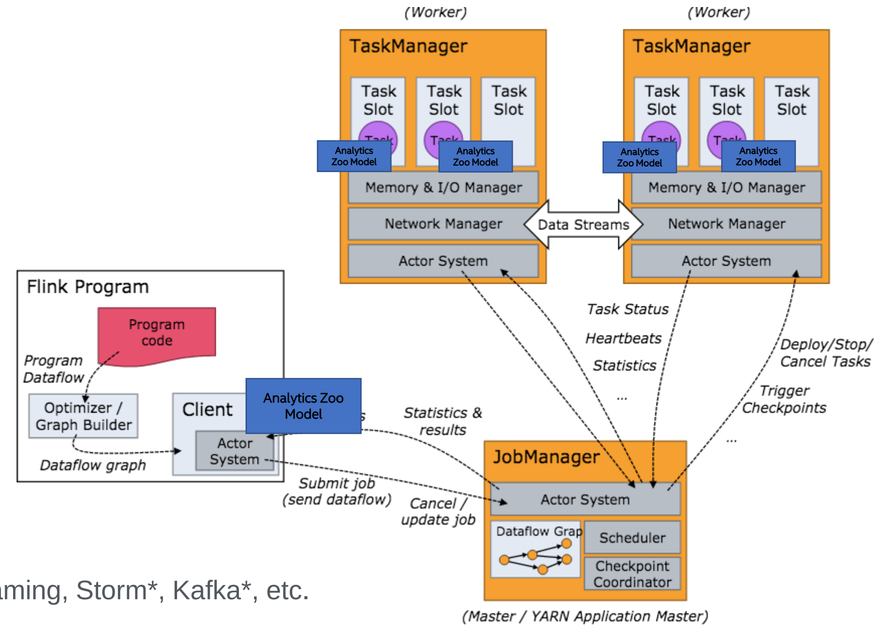
Distributed TensorFlow* Pipeline

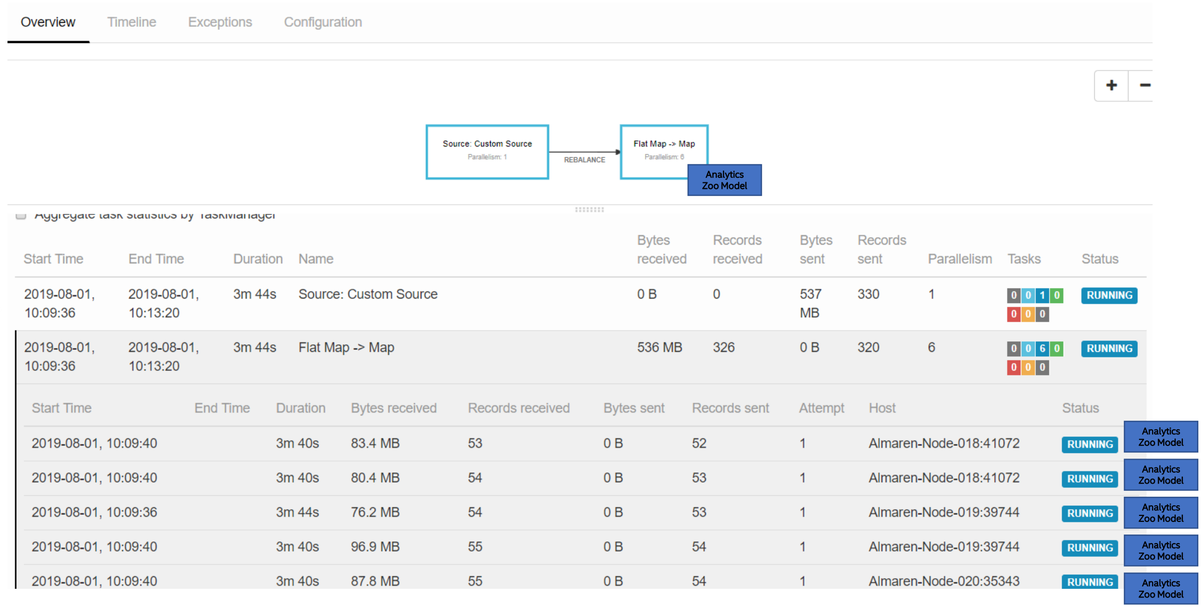


分布式、实时 (流式) 模型推理流水线

Distributed and Real time (streaming) Inference Pipeline

- 纯Java或Python API
- 支持Flink*, Spark* Streaming, Storm*, Kafka*等
- 支持Web Services
- 使用OpenVINO和DL Boost(VNNI) 加速



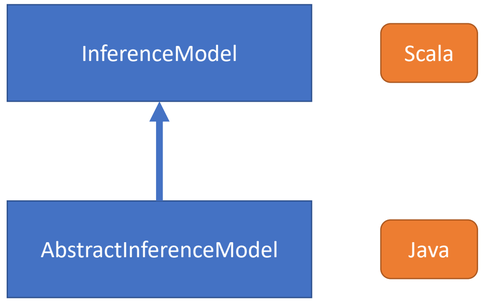


Pure Java, Python API
Support Flink*, Spark* Streaming, Storm*, Kafka*, etc.
Support Web Services
Accelerate with OpenVINO and DL Boost(VNNI)

POJO Style的Inference Model

POJO Style Inference Model

- 纯Java API, 不依赖于任何计算框架，不需要特别的上下文
- 可使用于单机Java/Scala程序，Web Serving，Cluster Serving包括批处理，流处理等场景
- 支持Flink*, Spark* Streaming, Storm*, Kafka*等



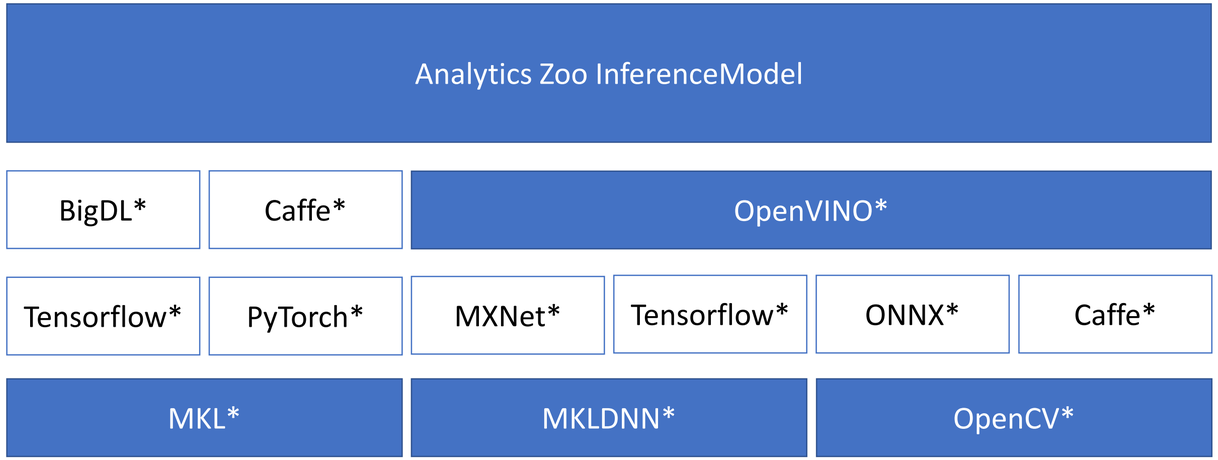

```
import com.intel.analytics.zoo.pipeline.inference.AbstractInferenceModel;

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

public class ServingExample {
  public static void main(String[] args) throws IOException {
    MyModel model = new MyModel();
    model.load(modelPath, weightPath);
    A data = …
    List<JTensor> inputs = preProcess(data);
    List<JTensor> outputs = model.predict(inputs);
    B results = postProcess(outputs);
  }
}
```

Pure Java API, requires no special computing framework or context
Support Java/Scala app, web serving and cluster serving.
Support Flink*, Spark* Streaming, Storm*, Kafka*, etc.

Inference Model 支持多种深度学习框架的模型

Inference Model supports lots of Deep Learning Frameworks

- 支持多种深度学习框架的模型
  - BigDL
  - Caffe*
  - Tensorflow*
  - PyTorch*
  - OpenVINO*
- 简单易用的API
  - 加载模型
    - laod
    - loadCaffe
    - loadTF
    - loadPyTorch
    - loadOpenVINO
  - 预测
    - predict

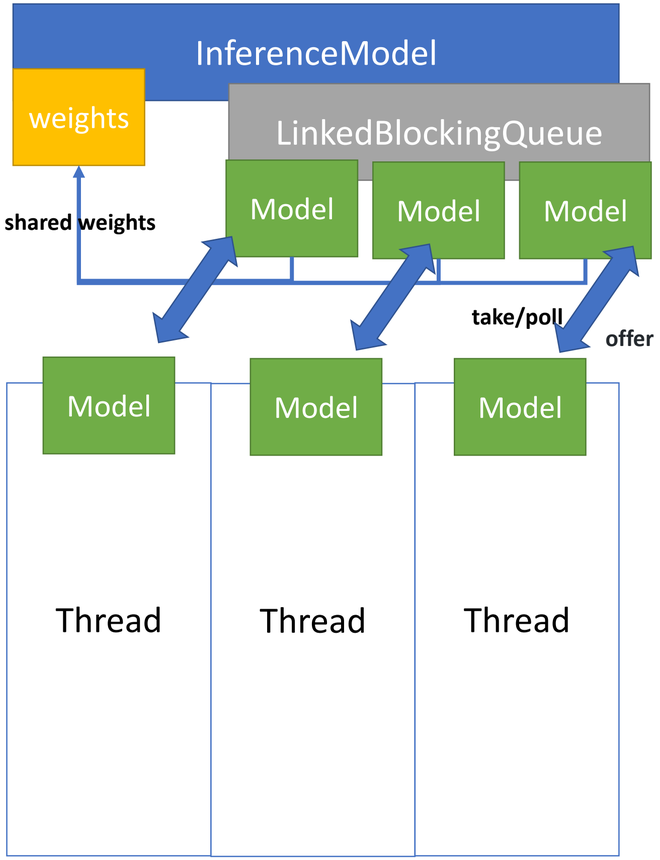


Support multiple Deep Learning frameworks.
Easy to use APIs to load model and do predict.

线程安全的Inference Model

Thread-Safe Inference Model

- 支持线程安全多模型
  - concurrentNum
    - model = modelQueue.take
  - autoScalingEnabled
    - model = modelQueue.poll()
    - model = this.originalModel.copy(1)(0)
- 多模型共享weights



Support thread-safe multiple working models.
Multiple working models share single copy of weights.

使用OpenVINO*加速模型推理

Model inference accelerating with OpenVINO*

- 支持Image Classification 和Object Detection等
- 支持加载TensorFlow*模型
- 支持模型动态Optimize及Calibrate
- 支持直接加载OpenVINO IR

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

sc = init_nncontext("OpenVINO Object Detection Inference Example")
images = ImageSet.read(options.img_path, sc,
   resize_height=600, resize_width=600).get_image().collect()
input_data = np.concatenate(
   [image.reshape((1, 1) + image.shape) for image in images], axis=0)

model = InferenceModel()
model.load_tf(options.model_path, backend="openvino",
model_type=options.model_type)
predictions = model.predict(input_data)

# Print the detection result of the first image.
print(predictions[0])
```

Support image classification and object detection
Support loading TensorFlow * models
Support direct loading of OpenVINO IR
Support model dynamic optimization and calibration

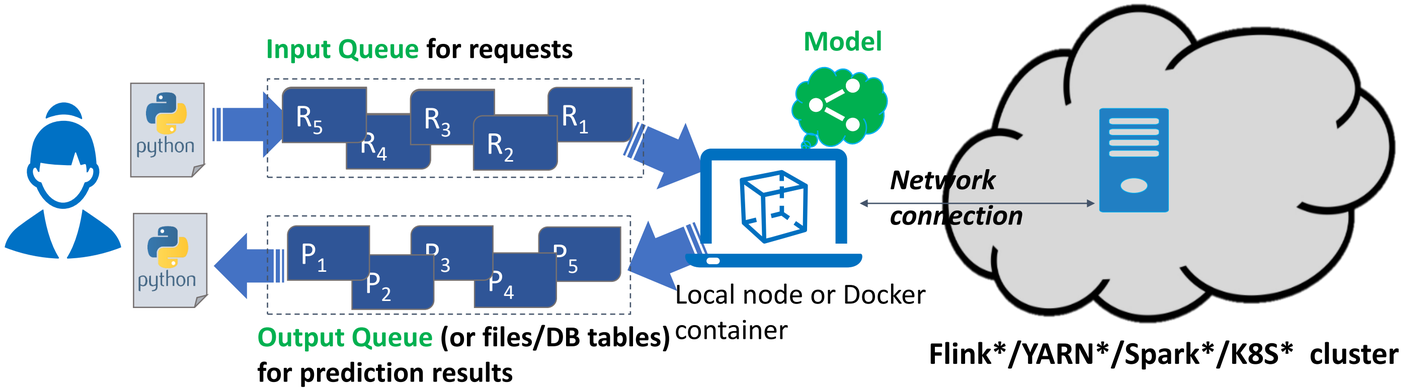
Analytics Zoo *Cluster Serving* 使分布式推理更加简单

Distributed Inference made easy with Analytics Zoo Cluster Serving

部署

√ 一个本地节点或者一个Docker容器

√ 已有的 Flink*/YARN*/Spark*/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

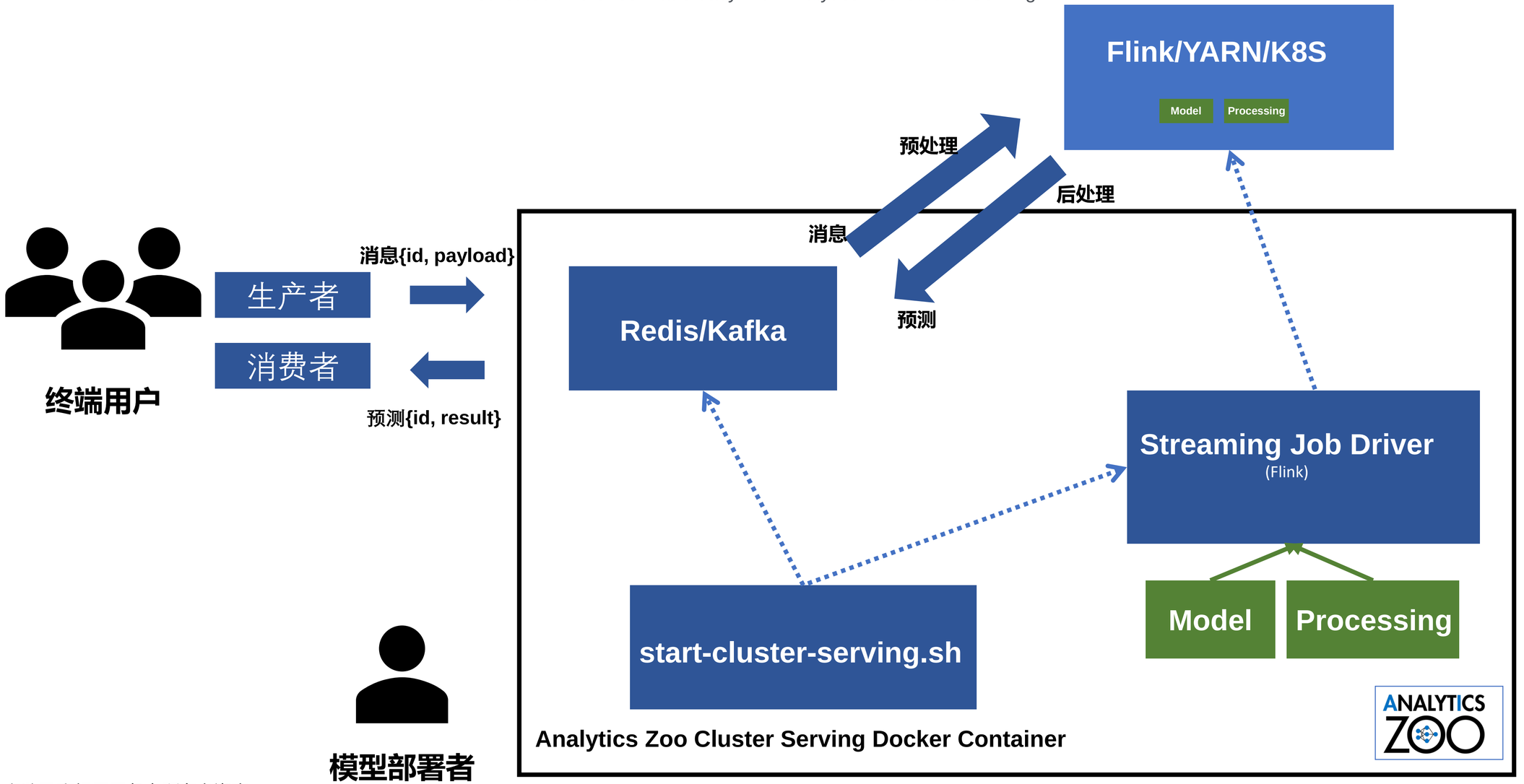
- 支持 TensorFlow*, Keras*, PyTorch*, Caffe*, BigDL 和 OpenVINO 的模型, 可使用 Int8 加速
- 通过Flink* 线性扩展

√ 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* 使分布式推理更加简单

Distributed Inference made easy with Analytics Zoo Cluster Serving



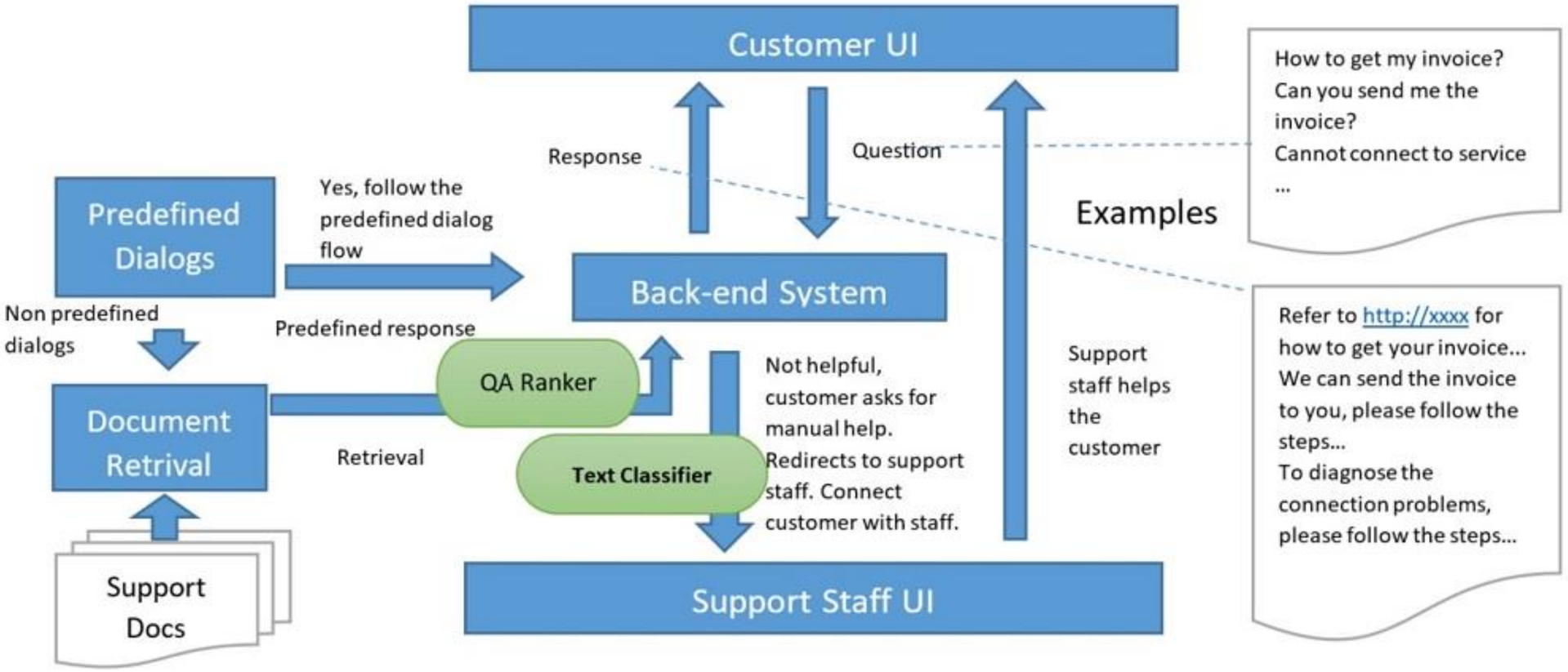
跨行业的端到端客户案例实践

Cross-industry End to End Use Cases

03

基于NLP的客户服务Chatbot for Microsoft Azure

NLP Based Customer Service Chatbot for Microsoft Azure



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/

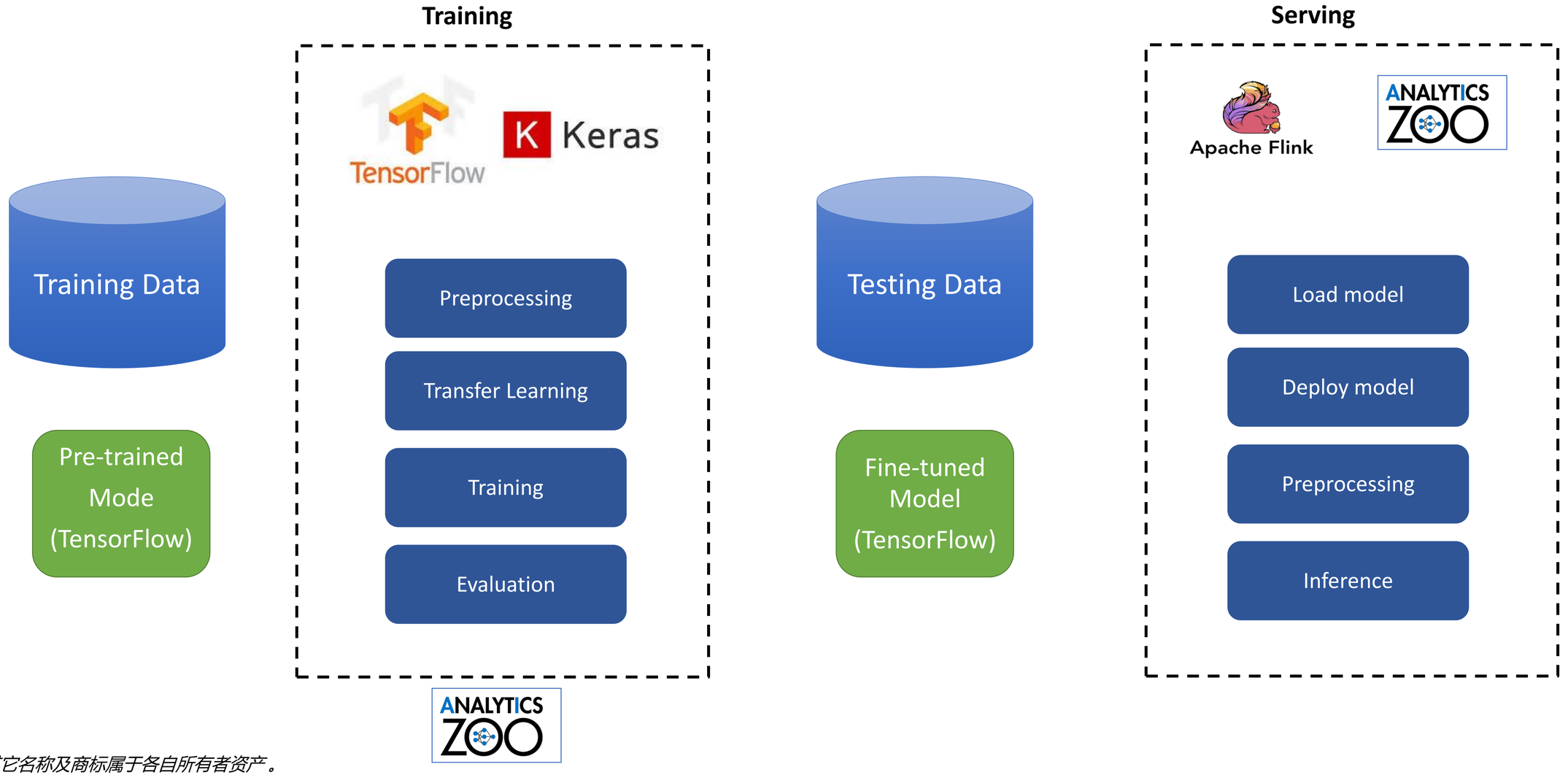
首届！Apache Flink 极客挑战赛强势来袭，重磅奖项等你拿，快来组队报名啦

Ververica 2019-07-24 17:51:26 浏览175

深度学习 大数据 性能优化 机器学习 性能 Apache 钉钉 开源大数据
流计算 大数据分析 ApacheFlink AI及大数据 实时技术

7 月 24 日，阿里云峰会上海开发者大会开源大数据专场，阿里巴巴集团副总裁、计算平台事业部总裁贾扬清与英特尔高级首席工程师、大数据分析和人工智能创新院院长戴金权共同发布首届 Apache Flink 极客挑战赛。

Apache Flink* 极客挑战赛垃圾图片分类



使用Analytics Zoo作迁移学习

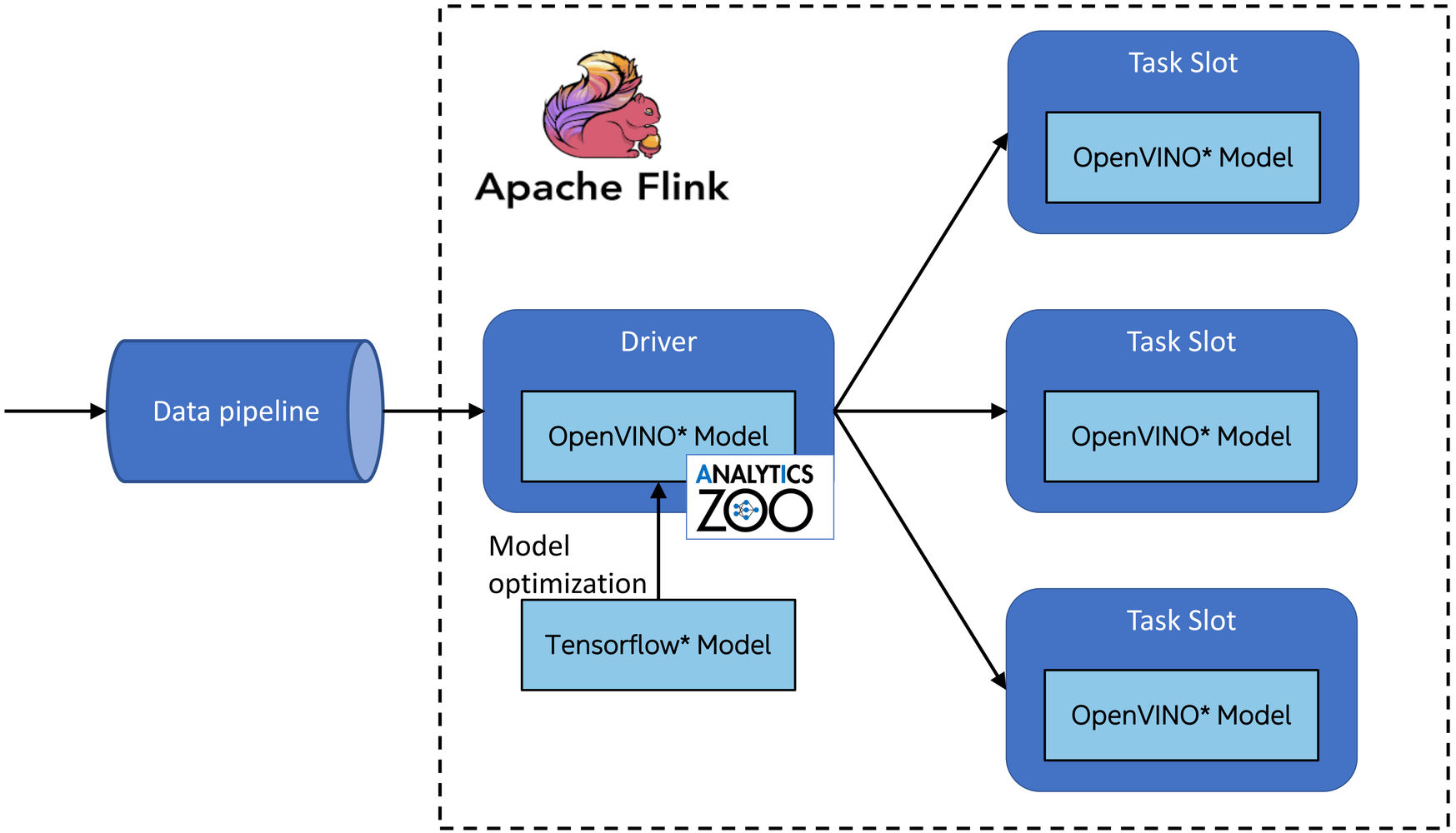
Transfer Learning with Analytics Zoo

- TFNet load TensorFlow* Saved Model
- Add extra layers
- Training with Estimator

```
val originalModel = TFNet.fromSavedModel(modelPath, inputs, outputs)

val model = Sequential[Float]()
model.add(originalModel)
model.add(new SpatialAveragePooling[Float](2, 2, globalPooling = true))
model.add(new Linear[Float](2048, 100))

val criterion = new CrossEntropyCriterion[Float]()
val adam = new Adam[Float]()
val validations = Array(new Top1Accuracy[Float], new Loss[Float])
val localEstimator = LocalEstimator(model, criterion, adam, validations,
threadNum)

val trainData = Cifar10DataLoader.loadTrainData(imageDirPath)
   .filter(_.label() <= 100).slice(0, 10 * batchSize)
val testData = Cifar10DataLoader.loadTestData(imageDirPath)
   .filter(_.label() <= 100).slice(0, 10 * batchSize)

localEstimator.fit(trainData, testData,
      ImageProcessing.labeledBGRImageToMiniBatchTransformer,
      batchSize, epoch)
```

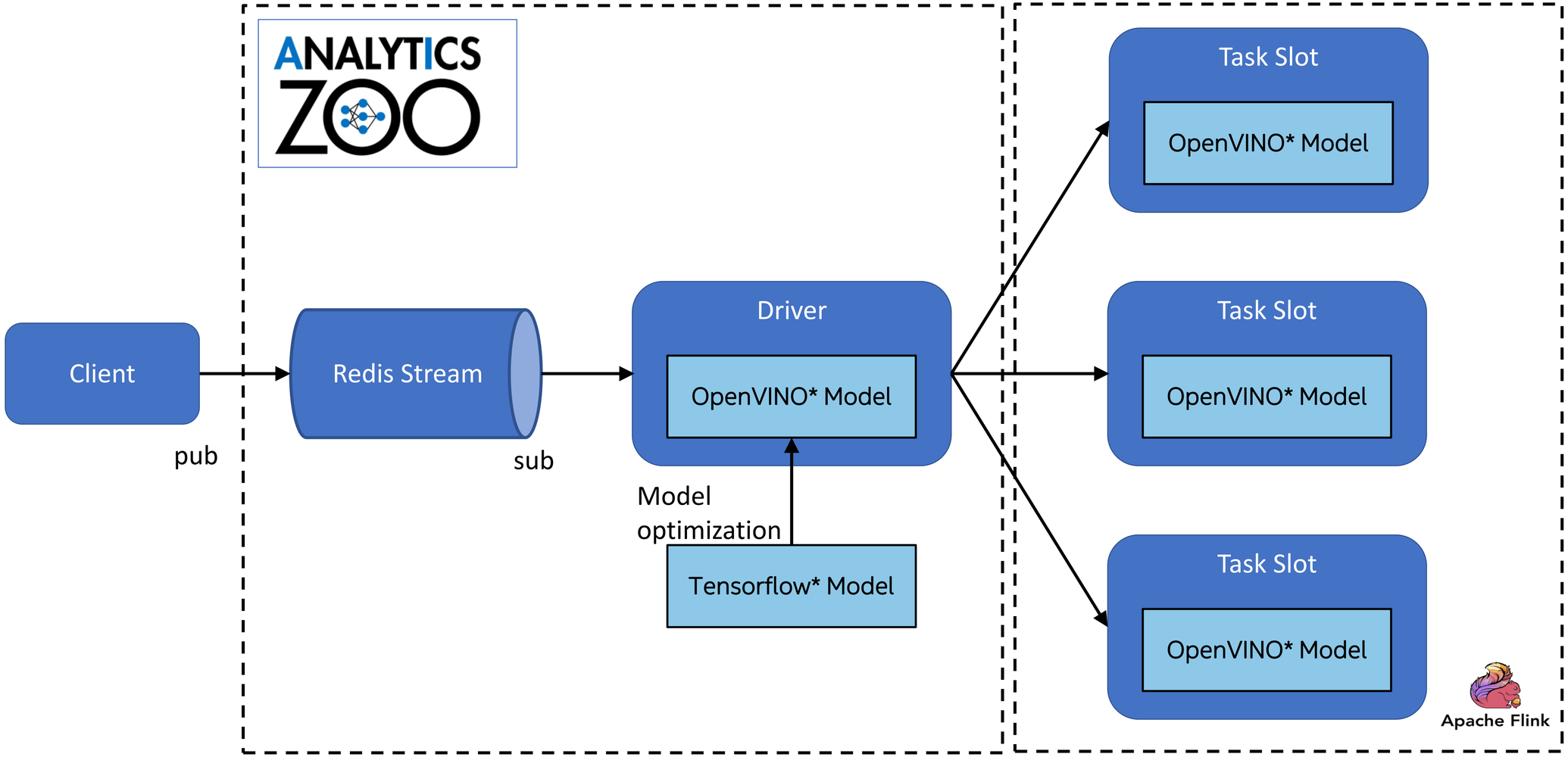
在Apache Flink*中使用Analytics Zoo进行分布式模型推理

Distributed Model Serving with Analytics Zoo in Apache Flink*



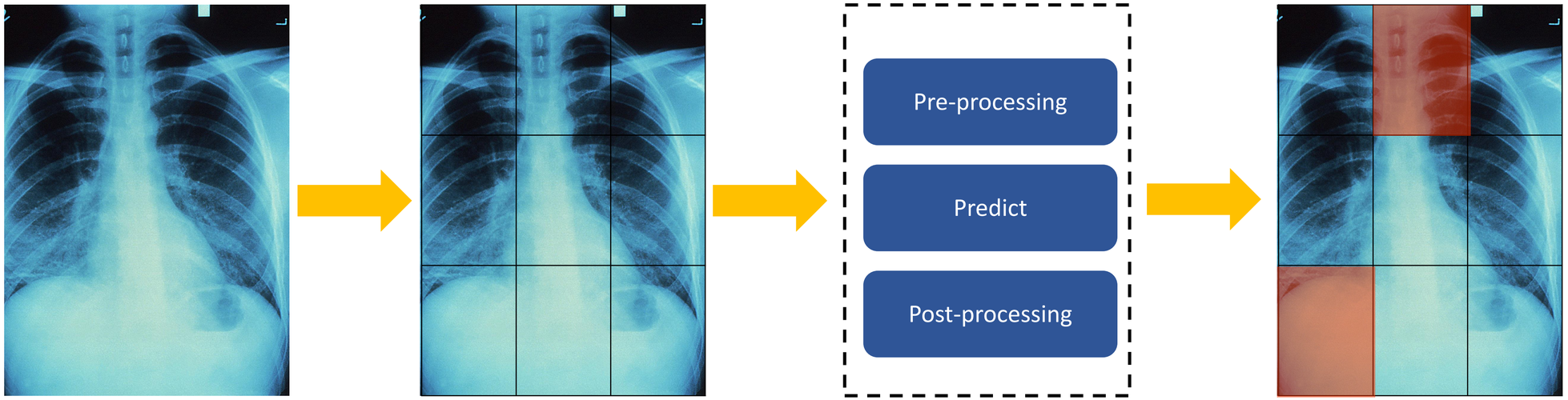
使用Analytics Zoo Cluster Serving进行分布式模型推理

Distributed Model Serving with Analytics Zoo Cluster Serving



使用Analytics Zoo Cluster Serving加速医疗影像分析

Accelerate medical image analysis with Analytics Zoo Cluster Serving



Raw image **10,000 images** **Pipeline logic** **Results**

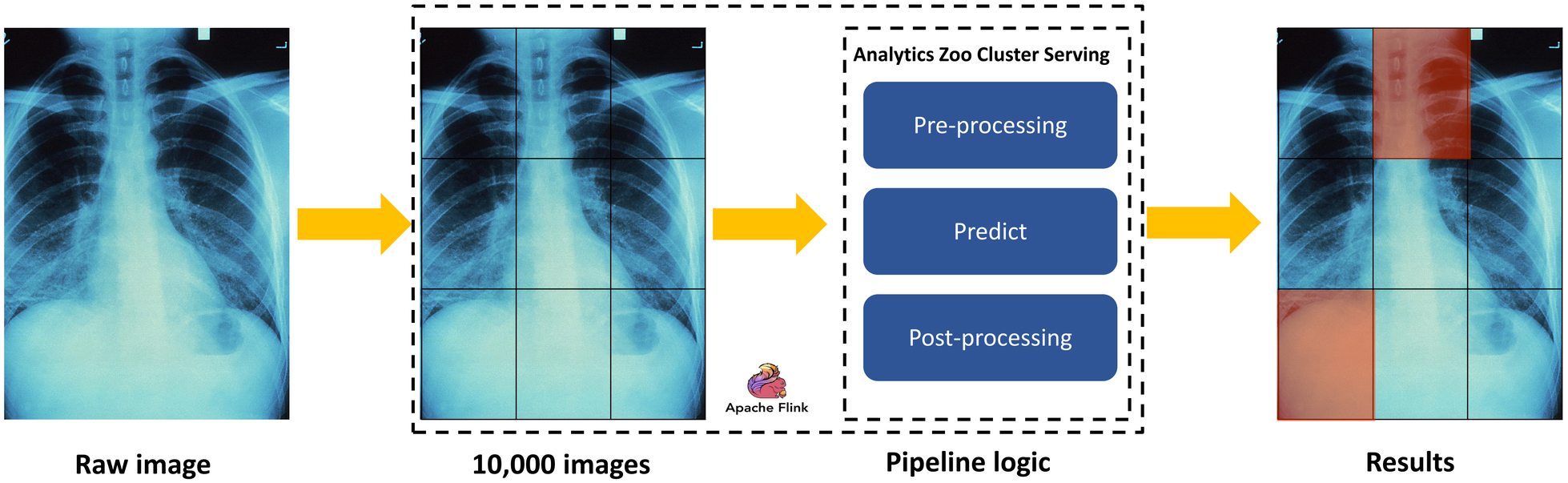
- 结果正确，但性能不可接受，每张原始图片需要1-2小时的处理与预测时间，很难扩展
- 性能瓶颈：预处理（split, crop, resize and normalization），推理

Unacceptable performance, 1-2 hours of processing and prediction, hard to scale
Performance bottlenecks: preprocessing (split, crop, resize and normalization)

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

使用Analytics Zoo Cluster Serving加速医疗影像分析

Accelerate medical image analysis with Analytics Zoo Cluster Serving



- 使用Analytics Zoo Cluster Serving, 处理时间：1-2小时→秒级
- 并发式图像处理使用Apache Flink*与Analytics Zoo(OpenCV*)
- 并发式模型推理使用Analytics Zoo(Caffe*，MKLDNN)
- 易于实现及扩展

With Analytics Zoo cluster serving, 1-2 hours → seconds
Parallel image processing using Apache Flink * and Analytics Zoo (OpenCV *)
Parallel model inference using Analytics Zoo (Caffe *, MKLDNN)
Easy to implement and scale

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

其他跨行业的端到端客户案例实践

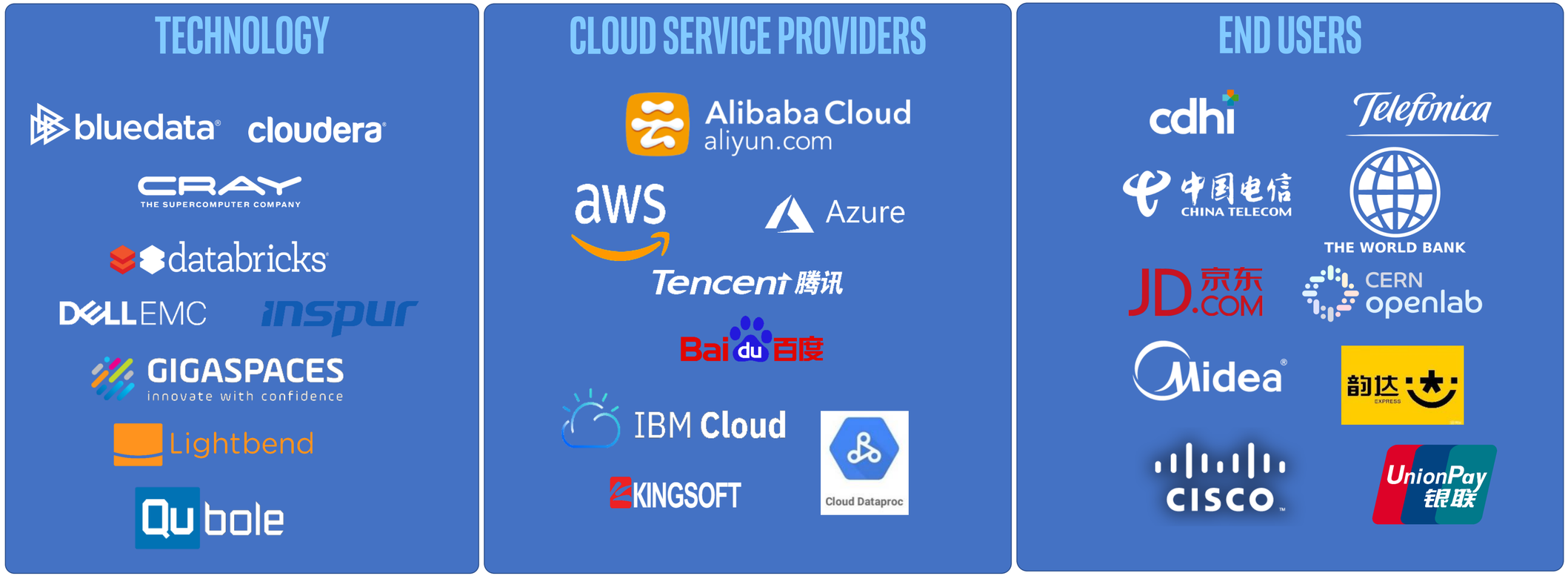
Other End to END Use Cases Examples

- 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
- 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
- CERN*：基于深度学习的高能物理粒子事件分类
  - 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

更多的案例实践

And Many More

Not a full list



software.intel.com/AIonBigData


THANKS
AI

LEGAL NOTICES AND DISCLAIMERS

- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations, and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit intel.com/performance.
- Intel does not control or audit the design or implementation of third-party benchmark data or websites referenced in this document. Intel encourages all of its customers to visit the referenced websites or others where similar performance benchmark data are reported and confirm whether the referenced benchmark data are accurate and reflect performance of systems available for purchase.
- Optimization notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.
- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com/benchmarks.
- Intel, the Intel logo, Intel Inside, the Intel Inside logo, Intel Atom, Intel Core, Iris, Movidius, Myriad, Intel Nervana, OpenVINO, Intel Optane, Stratix, and Xeon are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.
- *Other names and brands may be claimed as the property of others.
- © Intel Corporation

** 文中涉及的其它名称及商标属于各自所有者资产。*