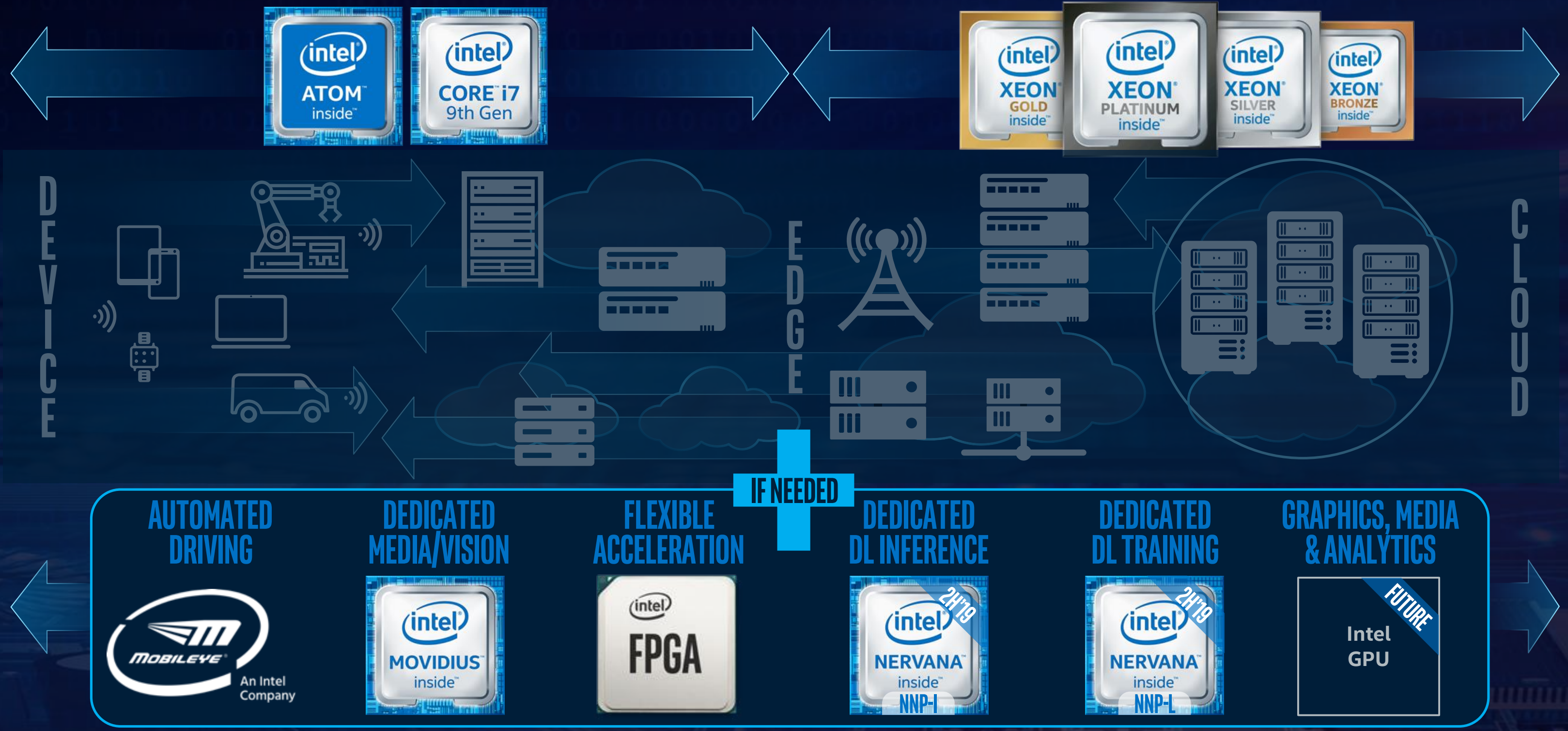
ANALYTICS ZOO: DISTRIBUTED TENSORFLOW AND KERAS ON APACHE SPARK

Yuhao Yang, Jiao Wang

Outline

- Analytics Zoo
  - Motivation
  - Architecture
  - Industry use cases
- Integrated examples:
  - Keras: Transfer Learning for image classification
  - Keras: Anomaly Detection
  - TensorFlow Training: Image Segmentation
  - PyTorch Inference: FaceGAN
  - PyTorch Training: Mnist

One Size Does Not Fit All



All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

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)

[1] An open source version is available at: 01.org/openvinotoolkit

*Other names and brands may be claimed as the property of others.

Developer personas show above represent the primary user base for each row, but are not mutually-exclusive

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

Real-World ML/DL Applications Are Complex Data Analytics Pipelines

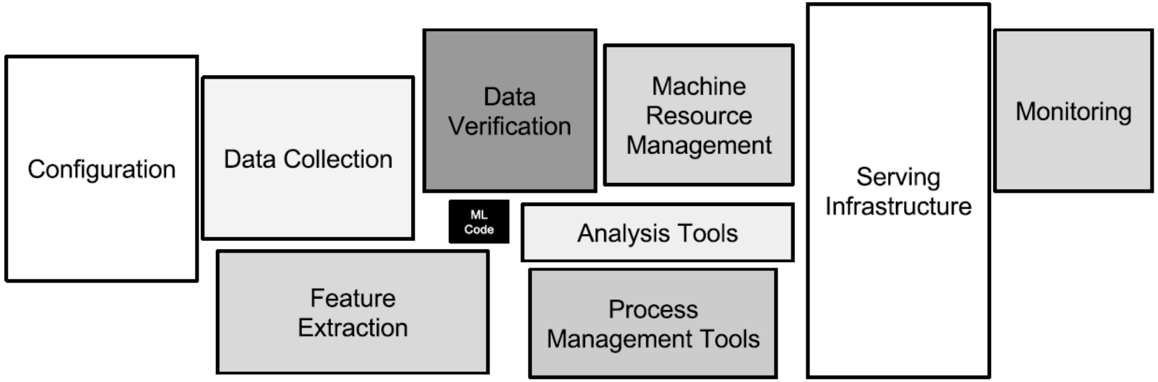
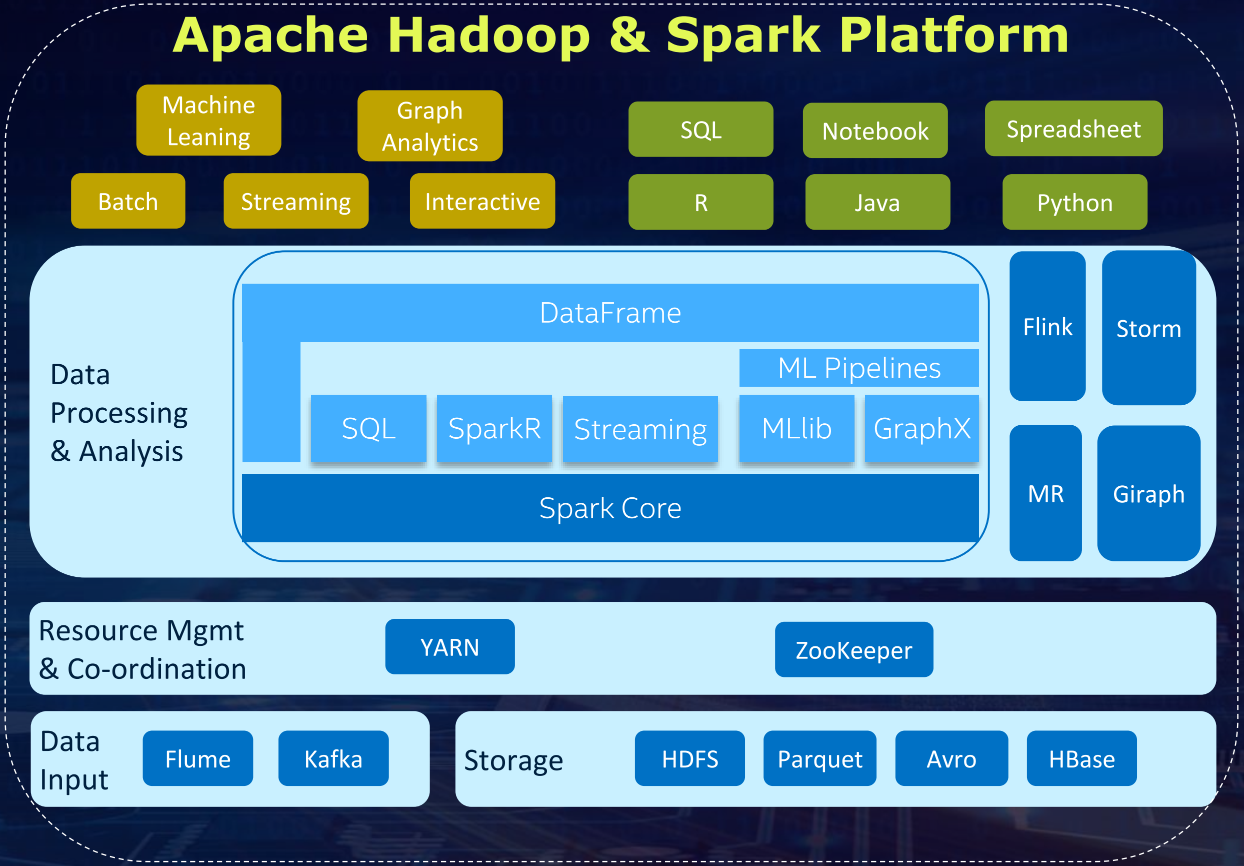


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

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

Unified Big Data Analytics Platform



Chasm b/w Deep Learning and Big Data Communities

Applications in JD.com

Large-scale image feature extraction

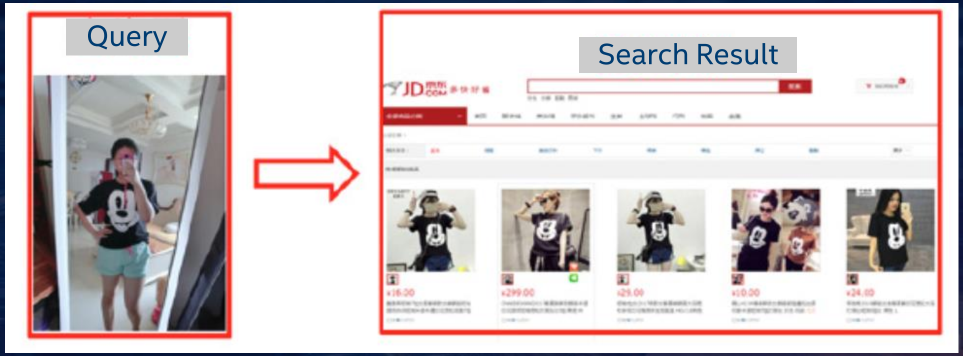
- Object detect (remove background, optional)
- Feature extraction

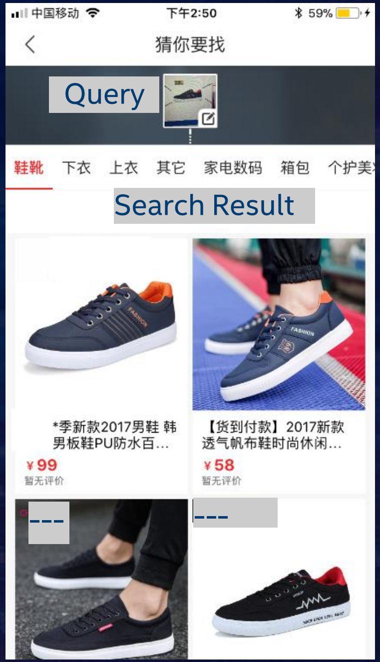
Application

- Similar image search
- Image Deduplication
    - Competitive price monitoring
    - IP (image copyright) protection system

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

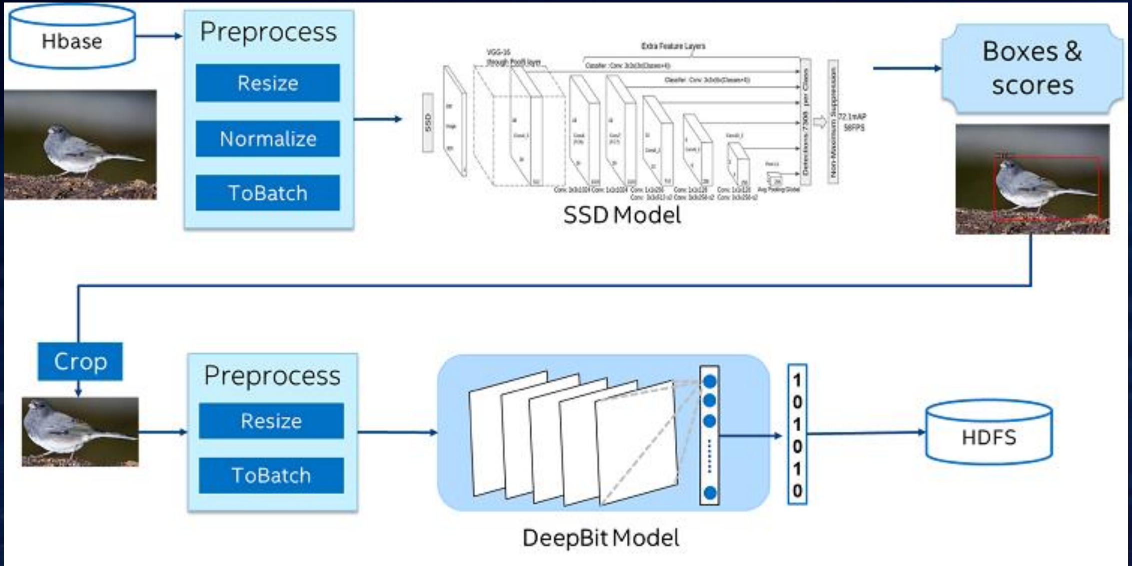
Similar Image Search

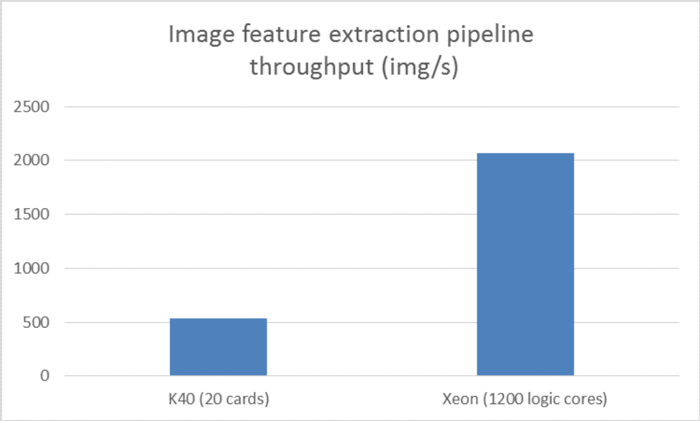




Source: “Bringing deep learning into big data analytics using BigDL”, Xianyan Jia and Zhenhua Wang, Strata Data Conference Singapore 2017

Production Deployment with Analytics Zoo for Spark and BigDL





- Reuse existing Hadoop/Spark clusters for deep learning with no changes (image search, IP protection, etc.)
- Efficiently scale out on Spark with superior performance (***3.83x*** speed-up vs. GPU severs) as benchmarked by JD

  http://mp.weixin.qq.com/s/xUCkzbHK4K06-v5qUsaNQQ
  https://software.intel.com/en-us/articles/building-large-scale-image-feature-extraction-with-bigdl-at-jdcom

AI on Apache Spark™

BigDL

Distributed, High-Performance
Deep Learning Framework
for Apache Spark*

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

Analytics Zoo

Analytics + AI Platform

Distributed TensorFlow*, Keras* and BigDL on Apache Spark*

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

Unifying Analytics + AI on Apache Spark*

Analytics Zoo: End-to-End DL Pipeline Made Easy for Big Data

Prototype on laptop using sample data

Experiment on clusters with history data

Deployment with production, distribtued big data pipelines

- **“Zero” code change from laptop to distributed cluster**
- **Directly accessing production big data (Hadoop/Hive/HBase)**
- **Easily prototyping the end-to-end pipeline**
- **Seamlessly deployed on production big data clusters**

What is Analytics Zoo?

Analytics Zoo

Unified Analytics + AI Platform for Big Data

Use case	Recommendation · Anomaly Detection · Text Classification · Text Matching
Model	Image Classification · Object Detection · Seq2Seq · Transformer · BERT
Feature Engineering	image · 3D image · text · Time series
High Level Pipelines	tfpark: Distributed TF on Spark · Distributed Keras w/ autograd on Spark · nnframes: Spark Dataframes & ML Pipelines for Deep Learning · Distributed Model Serving (batch, streaming & online)
Backend	TensorFlow* · Keras* · BigDL · OpenVINO · MKLDNN · Apache Spark* · Apache Flink*

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

Analytics Zoo Run as Standard Spark Programs

Standard Spark jobs

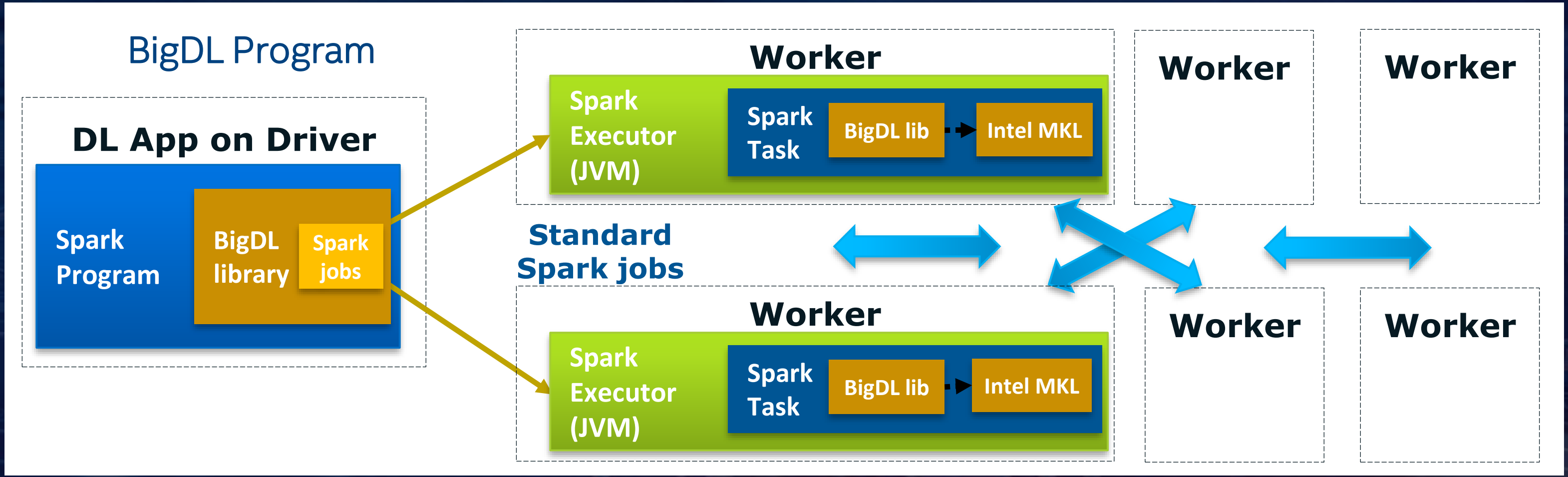
- No changes to the Spark or Hadoop clusters needed

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)



Analytics Zoo

Unified Analytics + AI Platform for Big Data

Build end-to-end deep learning applications for big data

- Distributed *TensorFlow* on Spark
- *Keras* API (with autograd & transfer learning support) on Spark
- *nnframes*: native DL support for Spark DataFrames and ML Pipelines

Productionize deep learning applications for big data at scale

- Plain Java/Python *model serving* APIs (w/ OpenVINO support)
- Support Web Services, Spark, Flink, Storm, Kafka, etc.

Out-of-the-box solutions

- Built-in deep learning *models*, *feature engineering* operations, and reference *use cases*

Distributed TF & Keras on Spark

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

Write TensorFlow code inline in PySpark program

```
def input_fn():
    #pyspark code
    training_rdd = spark.hadoopFile(...).map(...)
    dataset = TFDataset.from_rdd(training_rdd,
                                 features=(tf.float32, [28, 28, 1]),
                                 labels=(tf.int32, []),
                                 batch_size=320)
    return dataset

def model_fn(features, labels, mode):
    #tensorflow code
    import tensorflow as tf
    from nets import lenet
    slim = tf.contrib.slim
    with slim.arg_scope(lenet.lenet_arg_scope()):
        logits, end_points = lenet.lenet(features, num_classes=10,
                                         is_training=True)
    loss = tf.reduce_mean(
            tf.losses.sparse_softmax_cross_entropy(
            logits=logits, labels=labels))
    return TFEstimatorSpec(mode, predictions=logits, loss=loss)

#distributed training
estimator = TFEstimator(model_fn, tf.train.AdamOptimizer(),
                        model_dir="/tmp/estimator")
estimator.train(input_fn, steps=60000//320)
```

Spark Dataframe & ML Pipeline for DL

```
#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)
```

Distributed PyTorch on Spark

- Inference with Pre-trained models

- Training with PyTorch models and Loss functions
  1) Define model and Loss function with Pytorch
  2) Load and transform data with Spark and Analytics Zoo
  3) Train
  4) Save result to Torch script module.

- Combine with Spark ML pipeline

PyTorch Script

- A representation of a PyTorch model that can be understood, compiled and serialized by the Torch Script compiler

- C++ and Python
  - –AZ connected it with JVM

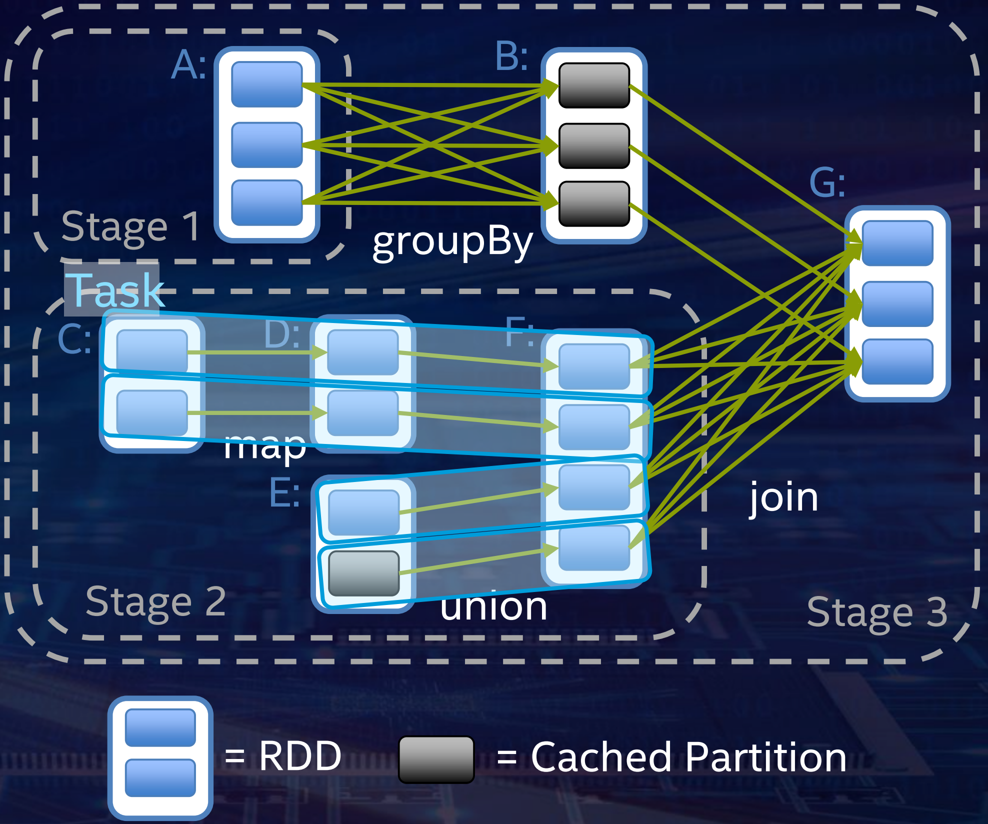
- Tracing + annotations

Distributed Training

Apache Spark

Spark compute model

- Data parallel
- Functional, coarse-grained operators
  - Immutable RDDs
  - Applying the same operation (e.g., `map`, `filter`, etc.) to all data items



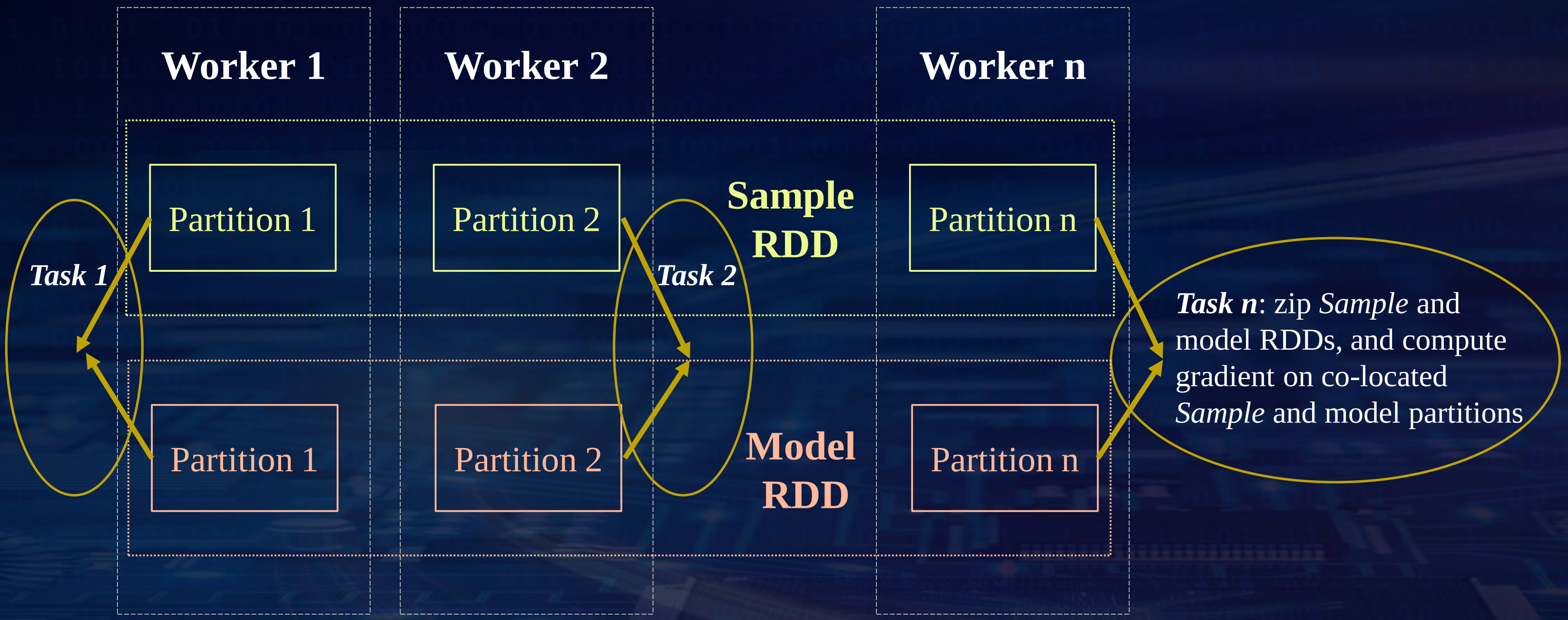
Source: "Parallel programming with Spark", Matei Zaharia, AMPCamp 3

Distributed Training in BigDL

Data Parallel, Synchronous Mini-Batch SGD

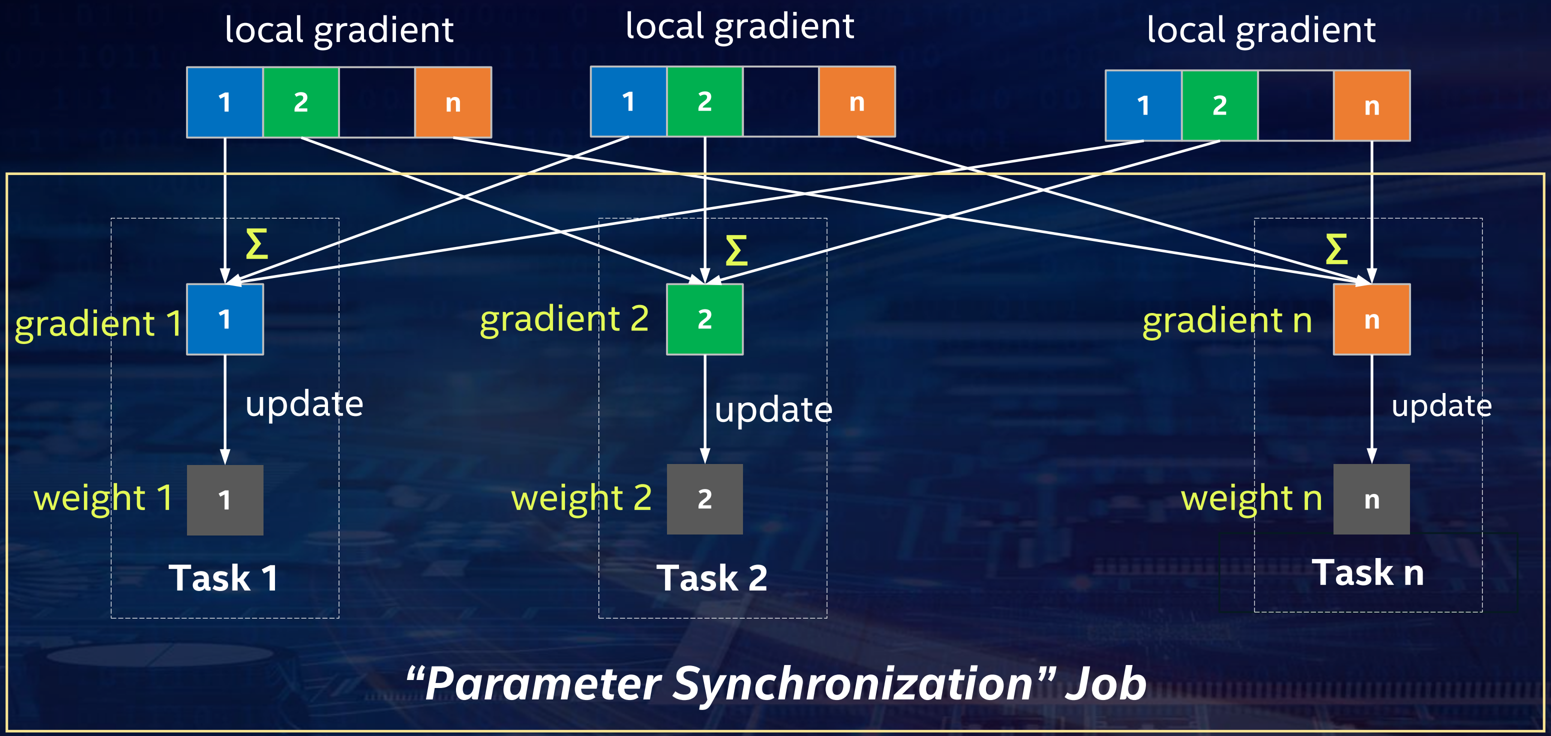
```
Prepare training data as an RDD of Samples
Construct an RDD of models (each being a replica of the original model)

for (i <- 1 to N) {
  //"model forward-backward" job
  for each task in the Spark job:
     read the latest weights
     get a random batch of data from local Sample partition
     compute errors (forward on local model replica)
     compute gradients (backward on local model replica)

  //"parameter synchronization" job
  aggregate (sum) all the gradients
  update the weights per specified optimization method
}
```

Data Parallel Training

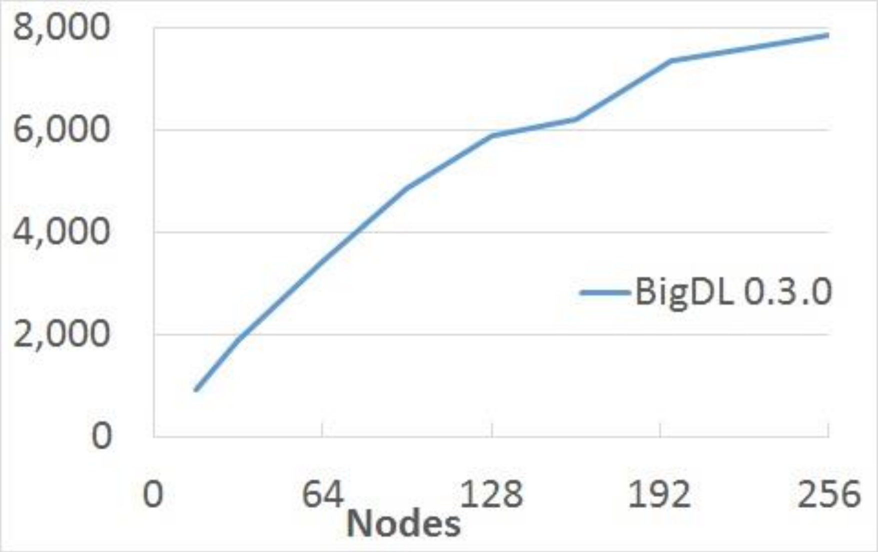


"Model Forward-Backward" Job

Parameter Synchronization



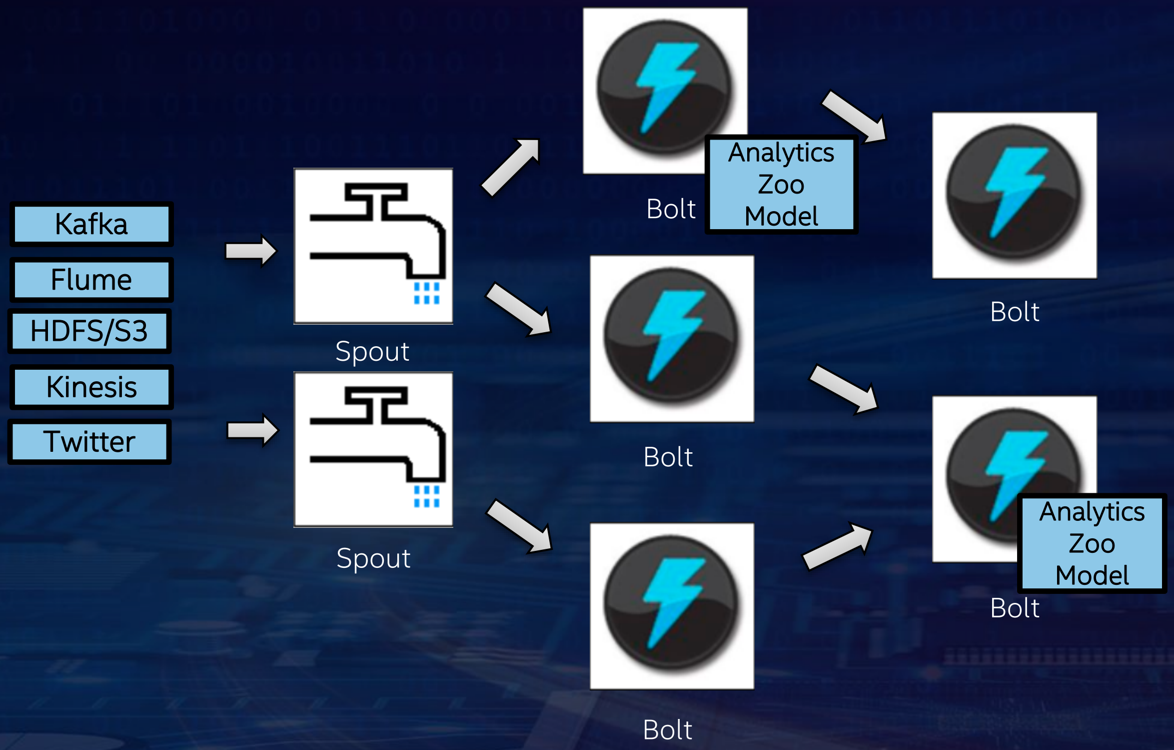
Training Scalability



Throughput of ImageNet Inception v1 training (w/ BigDL 0.3.0 and dual-socket Intel Broadwell 2.1 GHz); the throughput scales almost linear up to 128 nodes (and continue to scale reasonably up to 256 nodes).

Source: Scalable Deep Learning with BigDL on the Urika-XC Software Suite
(https://www.cray.com/blog/scalable-deep-learning-bigdl-urika-xc-software-suite/)

Distributed Model Serving

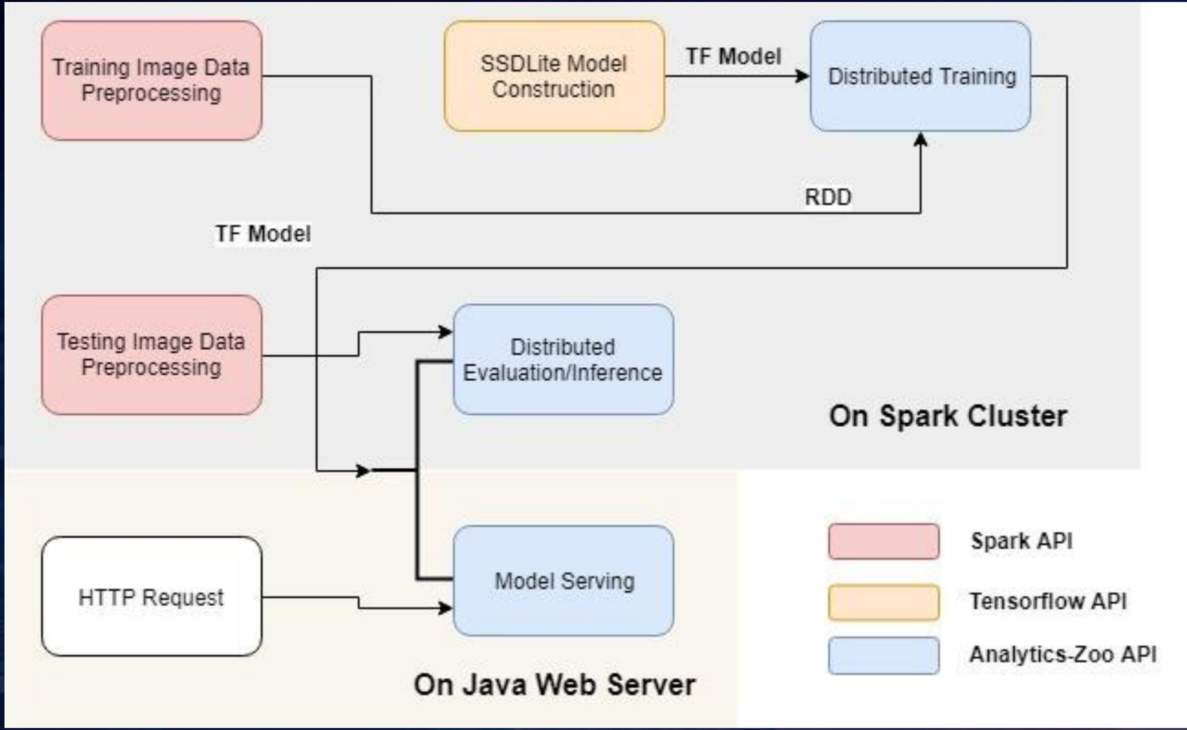


Distributed model serving in Web Service, Flink, Kafka, Storm, etc.

- **Plain Java or Python API, with OpenVINO and DL Boost (VNNI) support**

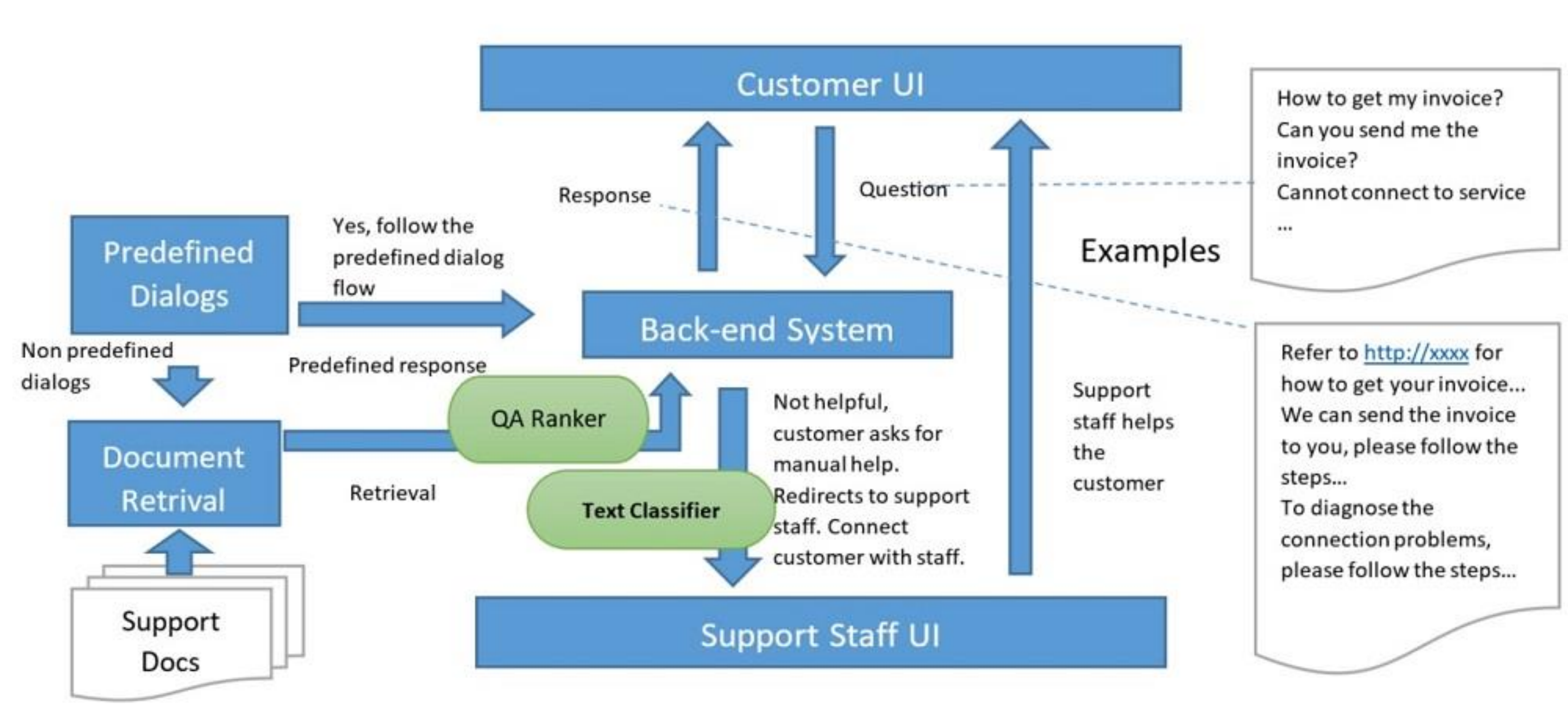
Analytics Zoo Use Cases

Computer Vision Based Product Defect Detection in Midea



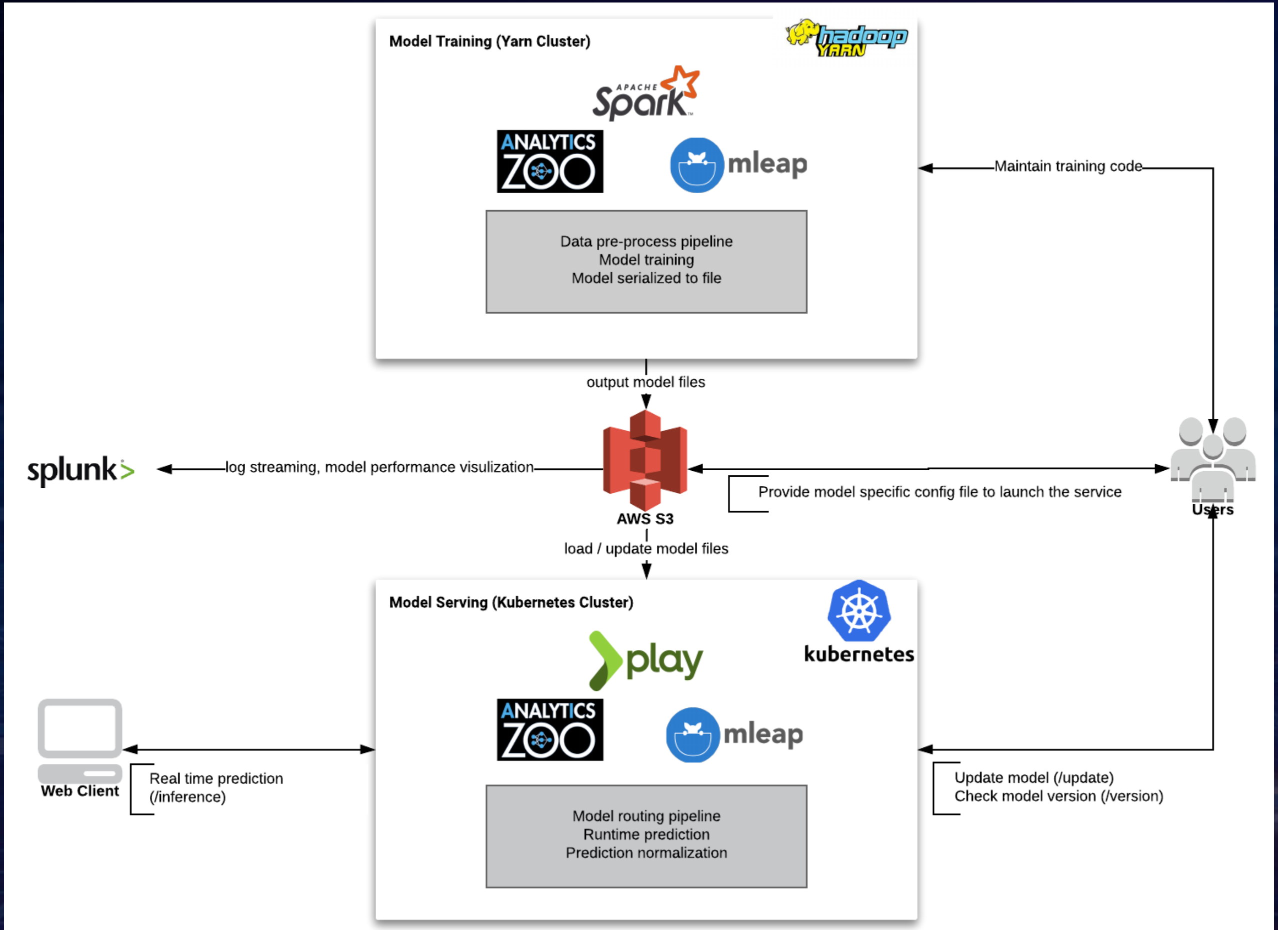
https://software.intel.com/en-us/articles/industrial-inspection-platform-in-midea-and-kuka-using-distributed-tensorflow-on-analytics

NLP Based Customer Service Chatbot for Microsoft Azure



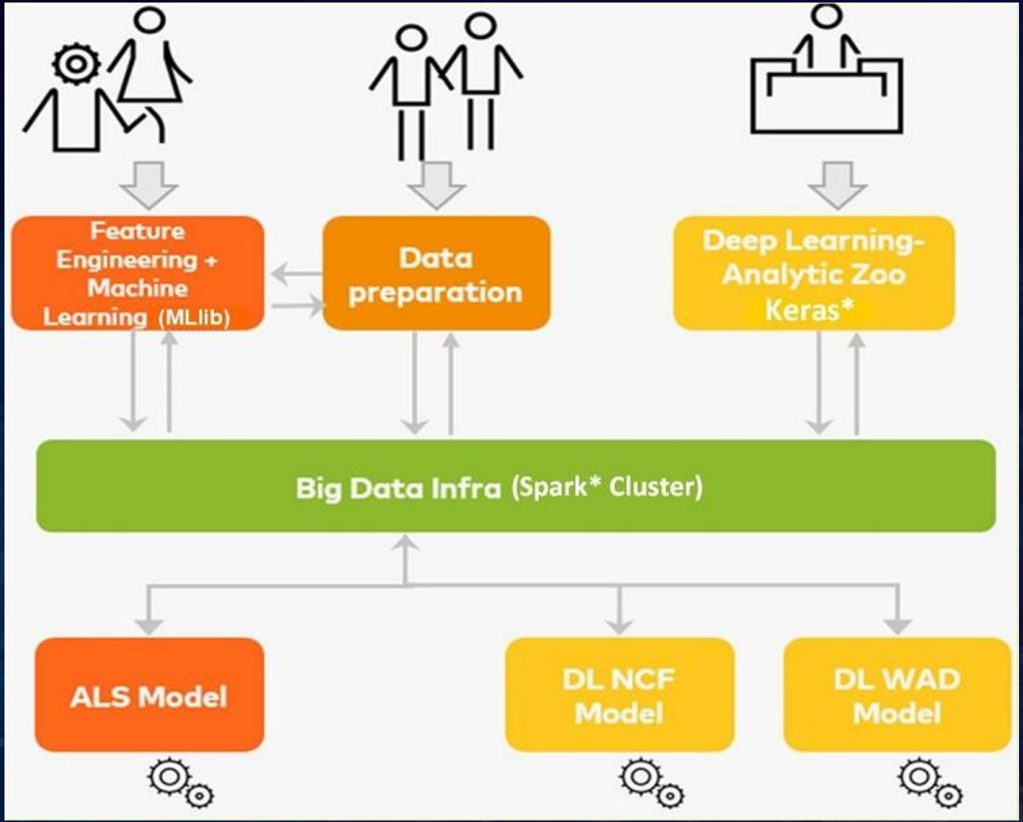
https://software.intel.com/en-us/articles/use-analytics-zoo-to-inject-ai-into-customer-service-platforms-on-microsoft-azure-part-1
https://www.infoq.com/articles/analytics-zoo-qa-module/

Product Recommendations in Office Depot



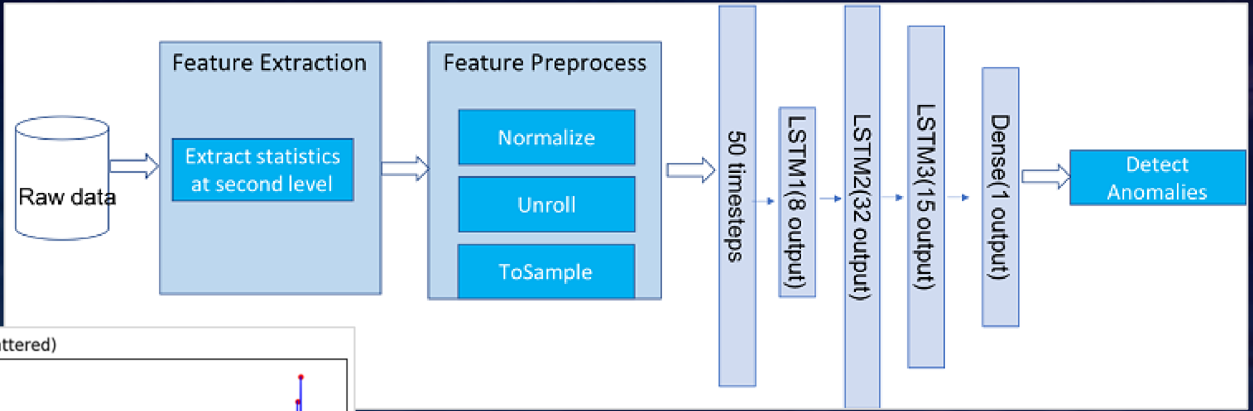
https://conferences.oreilly.com/strata/strata-ca-2019/public/schedule/detail/73079

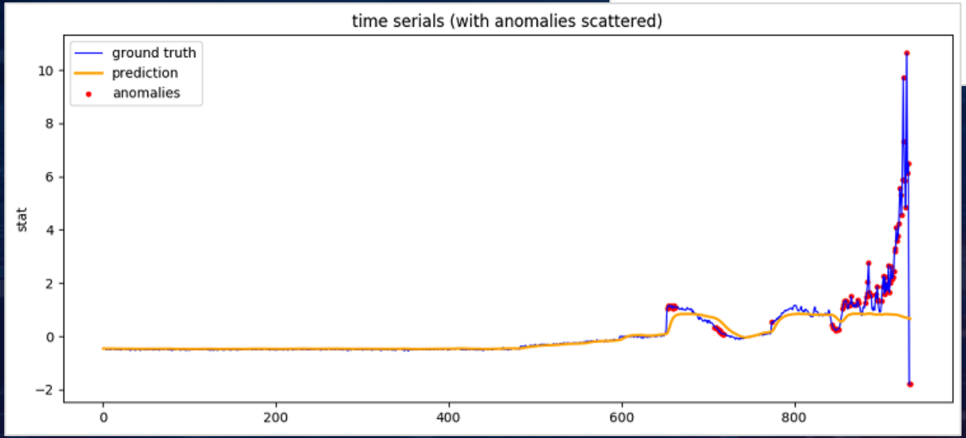
Recommender AI Service in MasterCard



https://software.intel.com/en-us/articles/deep-learning-with-analytic-zoo-optimizes-mastercard-recommender-ai-service

LSTM-Based Time Series Anomaly Detection for Baosight





https://software.intel.com/en-us/articles/lstm-based-time-series-anomaly-detection-using-analytics-zoo-for-apache-spark-and-bigdl

And Many More

TECHNOLOGY

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

CLOUD SERVICE PROVIDERS

Alibaba Cloud
aliyun.com
aws
Cloud Dataproc
Azure
IBM Cloud
Telefónica
KINGSOFT

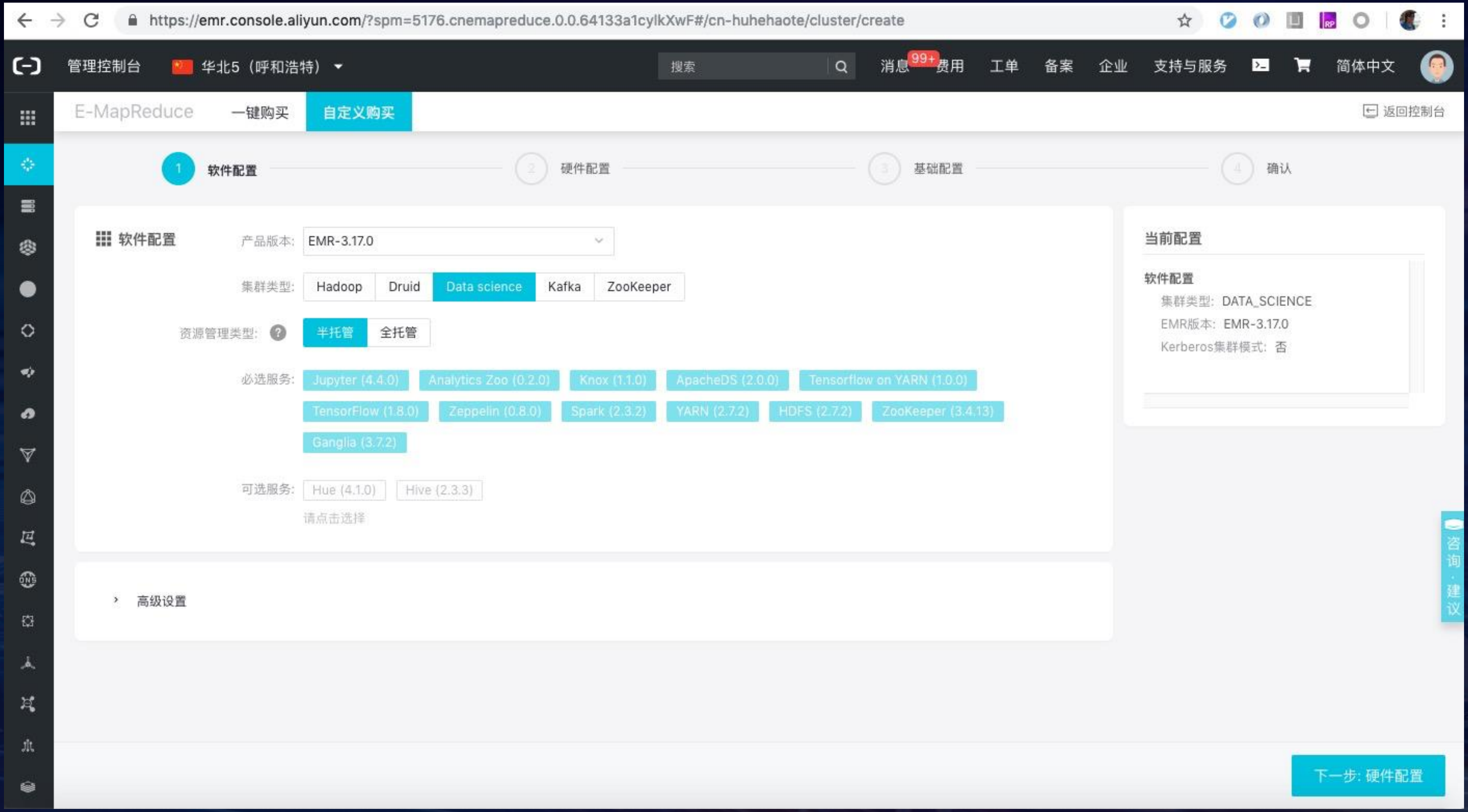
END USERS

cdhi
中国电信
CHINA TELECOM
JD.COM 京东
Midea®
UnionPay
银联

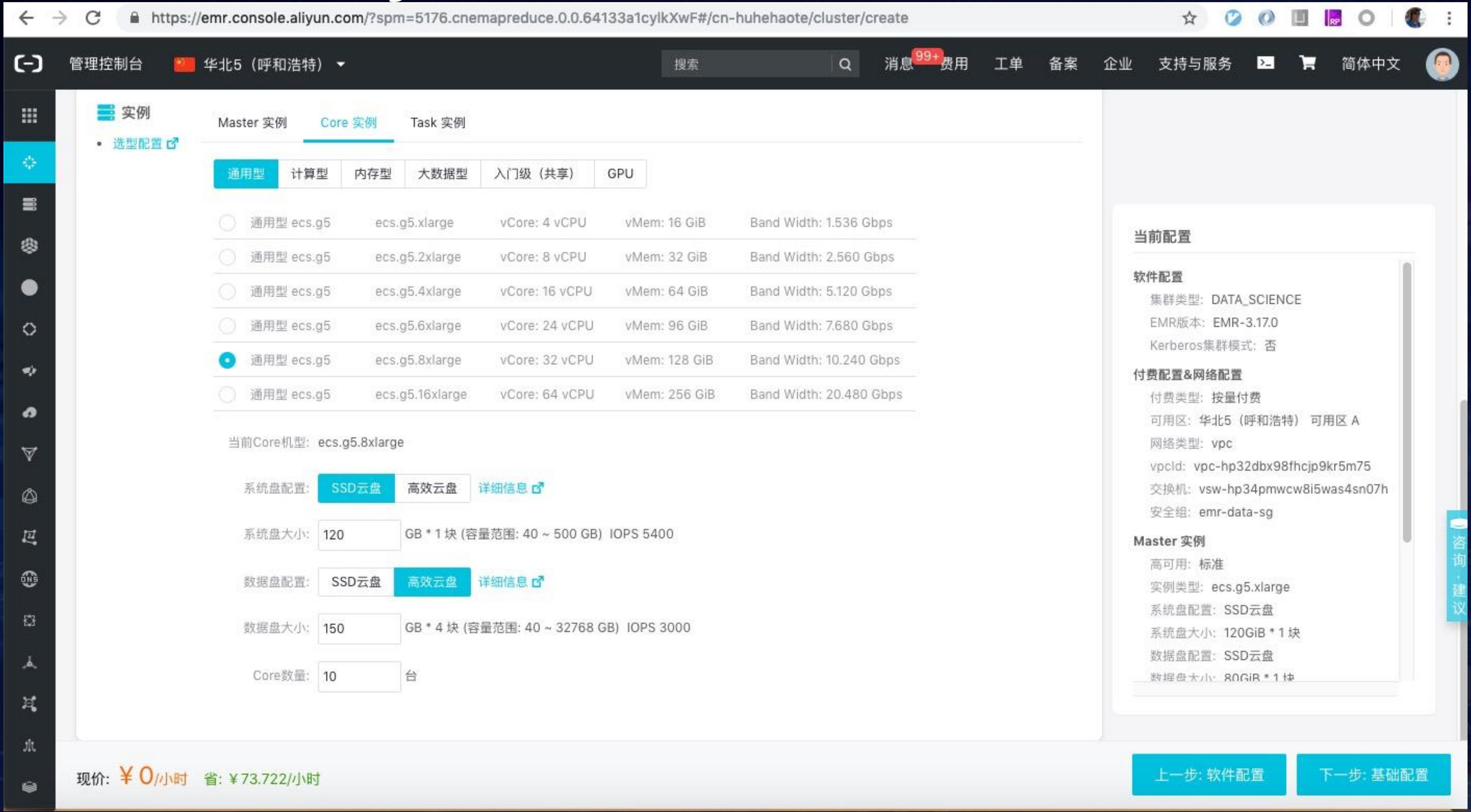
http://software.intel.com/bigdl/build

Not a full list

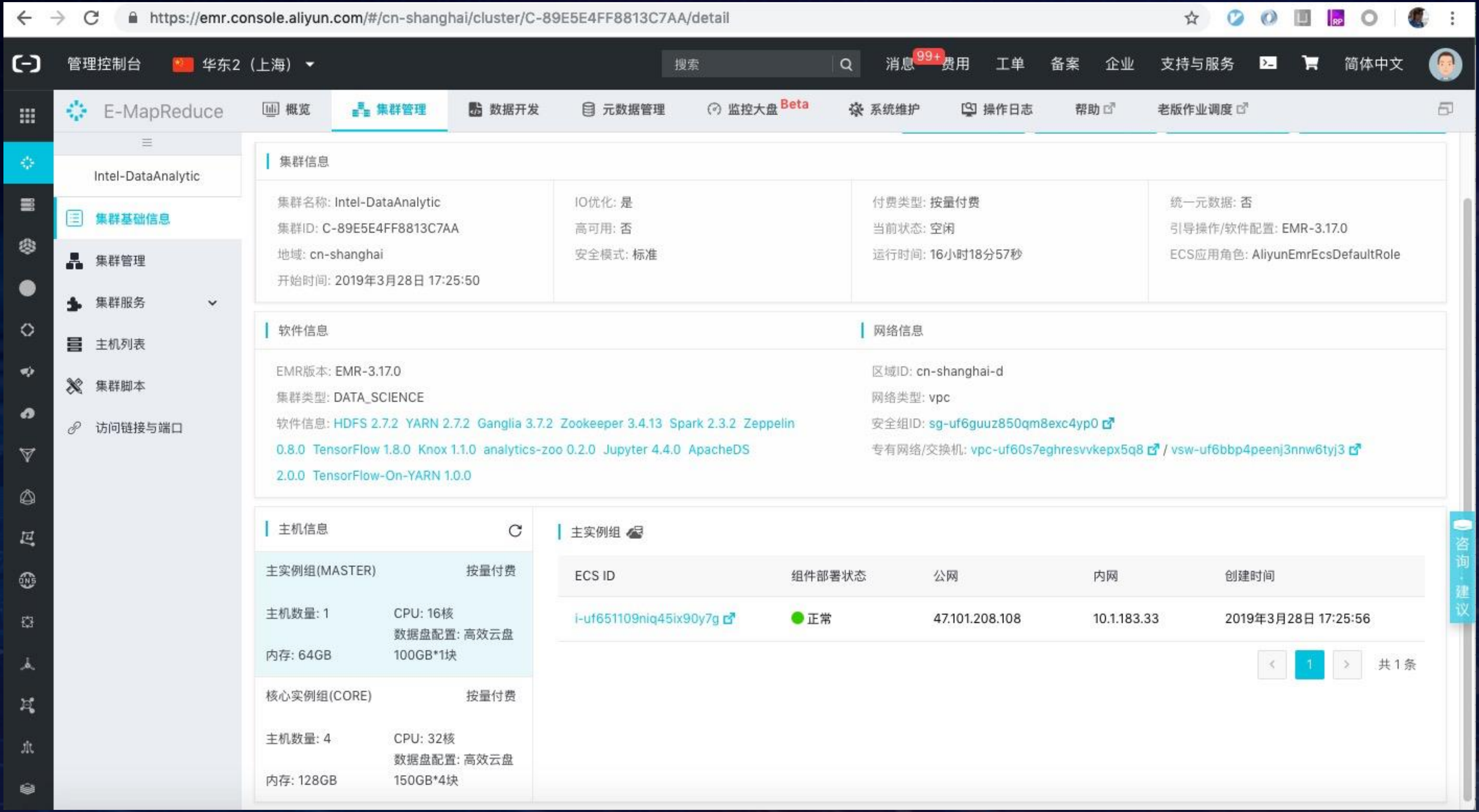
Analytics ZOO on AliYun EMR



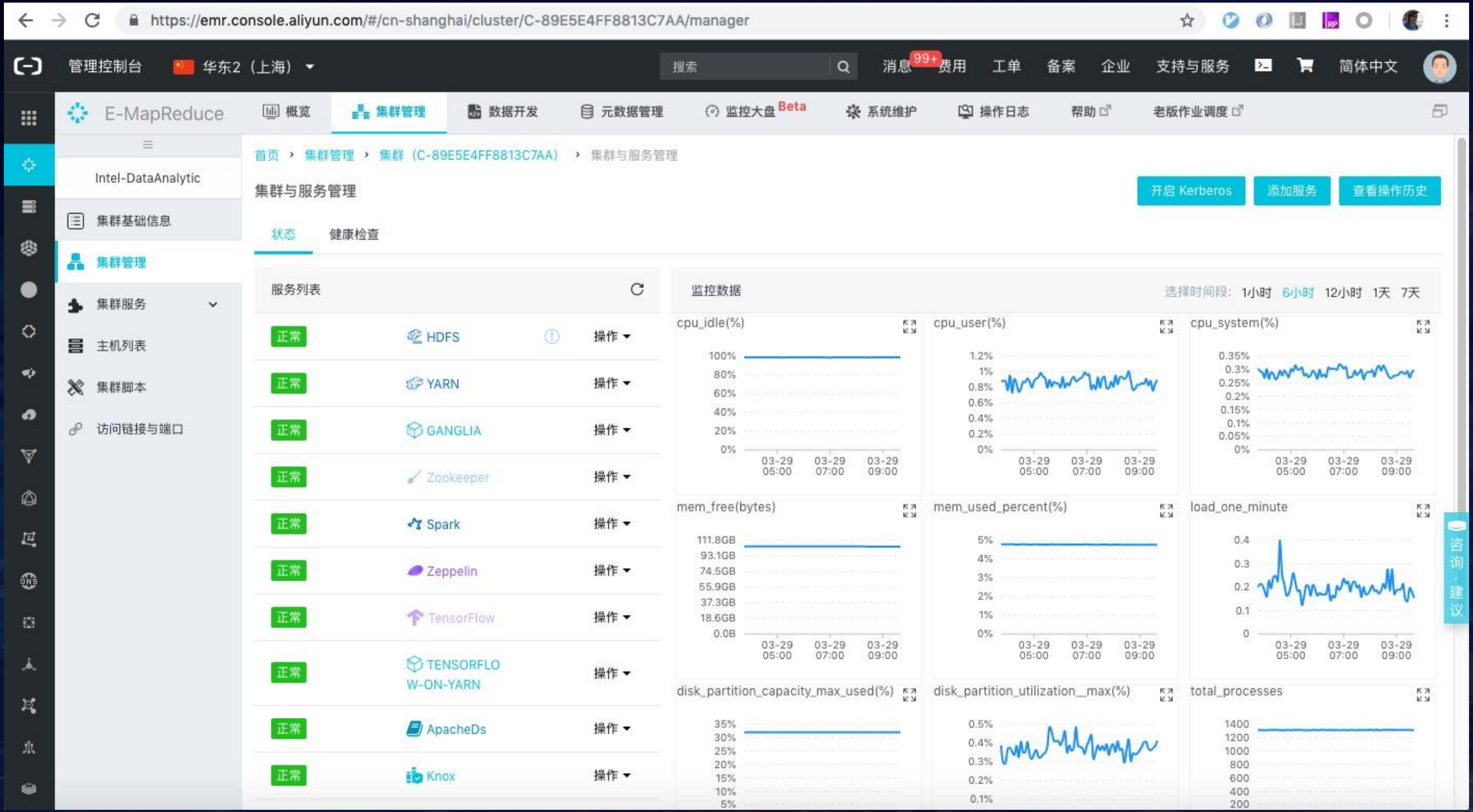
Analytics ZOO on AliYun EMR



Analytics ZOO on AliYun EMR



Analytics ZOO on AliYun EMR



Analytics ZOO on AliYun EMR

Stages for All Jobs

Active Stages: 1
Pending Stages: 1
Completed Stages: 698
Skipped Stages: 293

Active Stages (1)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input
1392	reduce at DistriOptimizer.scala:320 +details (kill)	2018/09/12 12:21:47	Unknown	0/2	

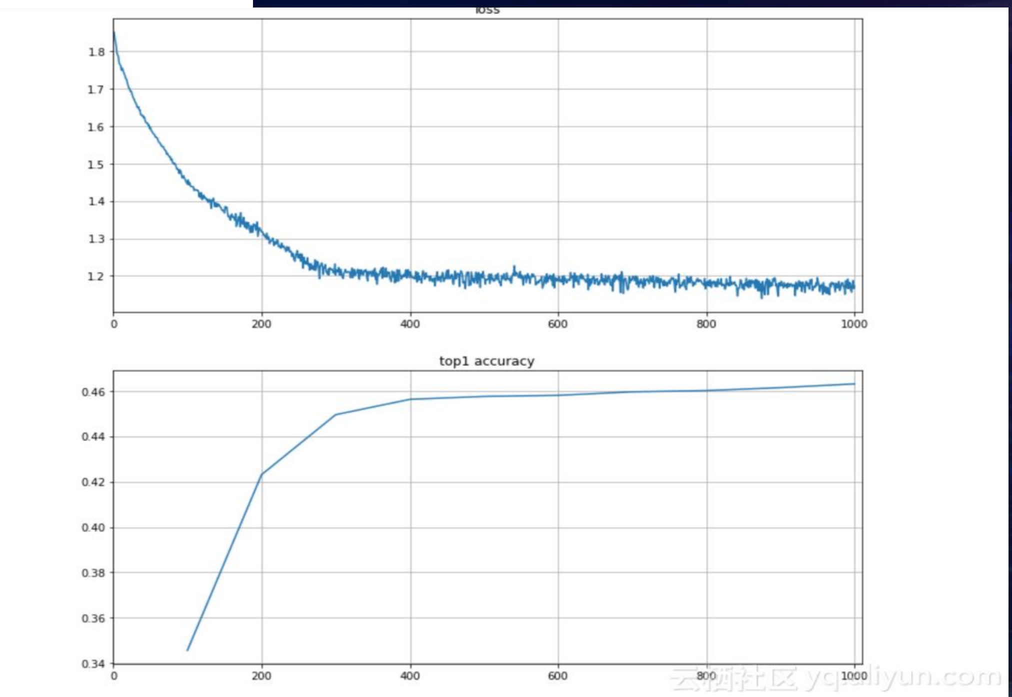
Pending Stages (1)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output
1391	coalesce at DataSet.scala:361 +details	Unknown	Unknown	0/4		

Completed Stages (698)

Page: 1 2 3 4 5 6 7 > 7 Pages. Jump to 1

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input
1390	count at DistriOptimizer.scala:369 +details	2018/09/12 12:21:47	12 ms	2/2	4.5 MB
1388	reduce at DistriOptimizer.scala:320 +details	2018/09/12 12:21:46	0.9 s	2/2	5.6 GB
1386	count at DistriOptimizer.scala:369 +details	2018/09/12 12:21:46	12 ms	2/2	4.5 MB
1384	reduce at DistriOptimizer.scala:320 +details	2018/09/12 12:21:45	1.0 s	2/2	5.6 GB
1382	count at DistriOptimizer.scala:369 +details	2018/09/12 12:21:45	11 ms	2/2	4.5 MB
1380	reduce at DistriOptimizer.scala:320 +details	2018/09/12 12:21:44	0.9 s	2/2	5.6 GB
1378	count at DistriOptimizer.scala:369 +details	2018/09/12 12:21:44	11 ms	2/2	4.5 MB
1376	reduce at DistriOptimizer.scala:320 +details	2018/09/12 12:21:43	1.0 s	2/2	5.6 GB
1374	count at DistriOptimizer.scala:369 +details	2018/09/12 12:21:43	11 ms	2/2	4.5 MB



Upcoming Analytics Zoo 0.6 Release

- Distributed PyTorch on Spark
- Ray on Spark
  - Run Ray programs directly on standard Hadoop/YARN clusters
- AutoML support
  - Automatic feature generation, model selection and hyper-parameter tuning for *time series prediction*
- Cluster serving
  - Distributed, real-time (streaming) model serving with simple pub-sub interface

Deep Learning Made Easy for Big Data

ANALYTICS ZOO

Unified Analytics + AI Platform

Distributed TensorFlow*, Keras* and BigDL on Apache Spark*

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

Colab

- Google colab
- Sign in with a Google account
- All notebooks available on
- https://github.com/intel-analytics/OreillyAI2019
- In Colab, open Notebook from Github, e.g. https://github.com/intel-analytics/OreillyAI2019/blob/master/keras/transfer_learning.ipynb

LEGAL 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**.