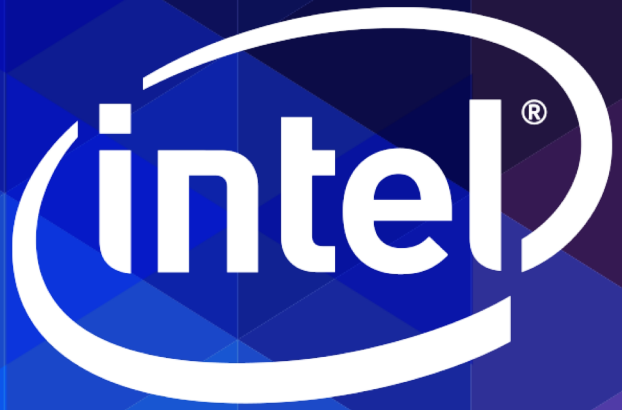


AI DEVCON 2018

AI DEVCON 2018

BUILD DEEP LEARNING POWERED BIG DATA SOLUTIONS WITH BIGDL FOR APACHE SPARK*

Sujee Maniyam

Big Data Practitioner and Co-founder, Elephant Scale

Brien Porter

Sr. Program Manager Big Data Technology, Intel Software

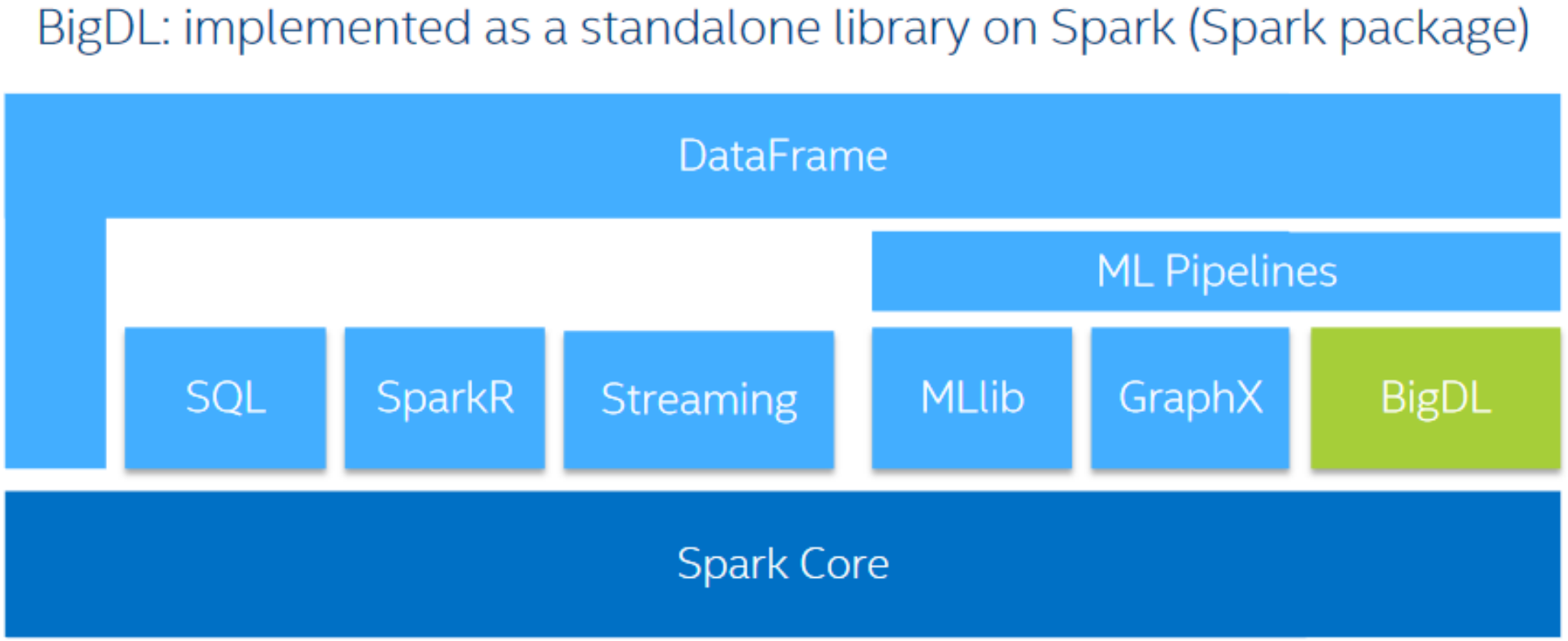
AGENDA

- Brief Overview of BigDL
- Hands-on Segment
  - Getting Started with BigDL Docker/VM
  - Building your first RNN with BigDL
  - Building a LSTM with BigDL
- Where can you access BigDL on AWS MP

BIGDL OVERVIEW

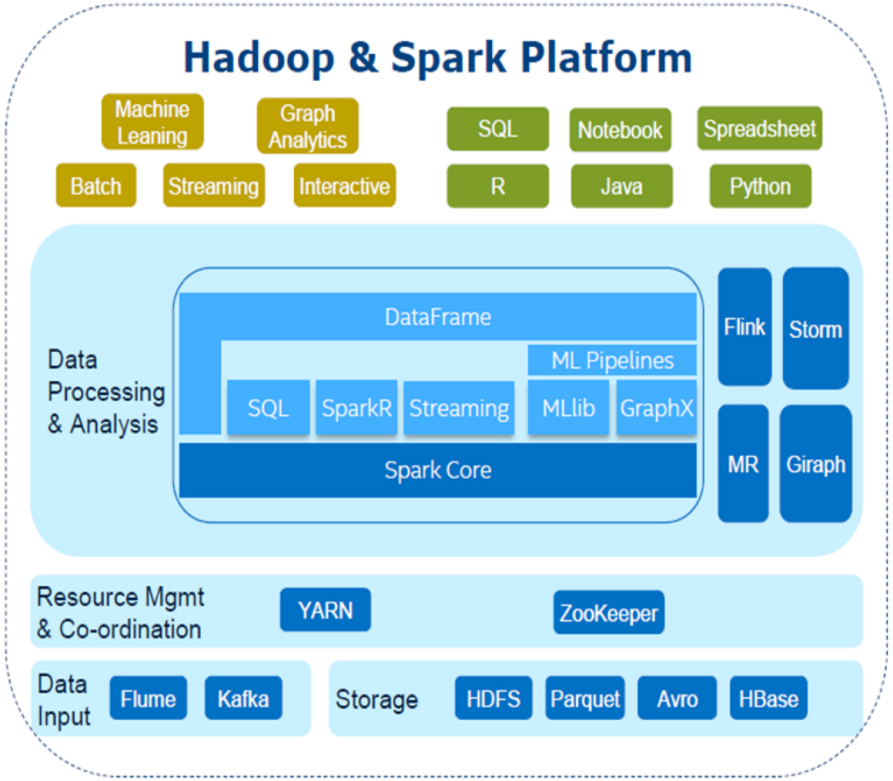
WHAT IS BIGDL?

- BigDL is a distributed deep learning library for Apache Spark*



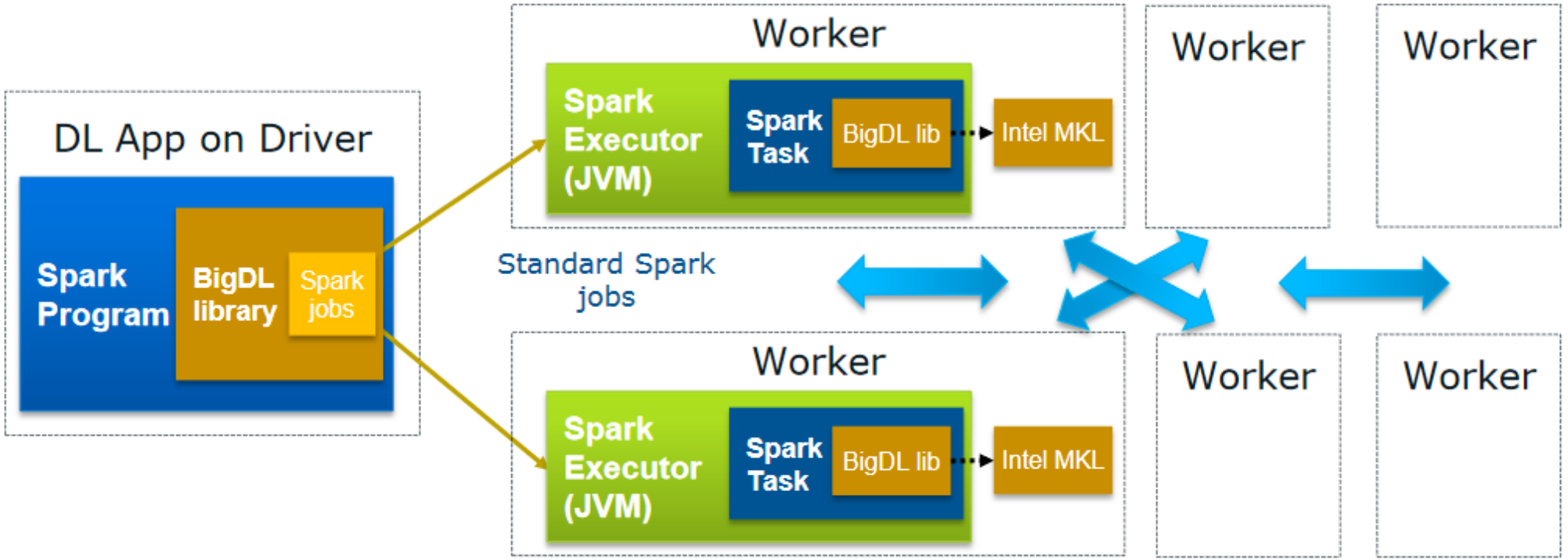
BIGDL IS DESIGNED FOR BIG DATA

- 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
  - Powered by Intel MKL and multi-threaded programming
- Efficient scale-out
  - Leveraging Spark for distributed training & inference



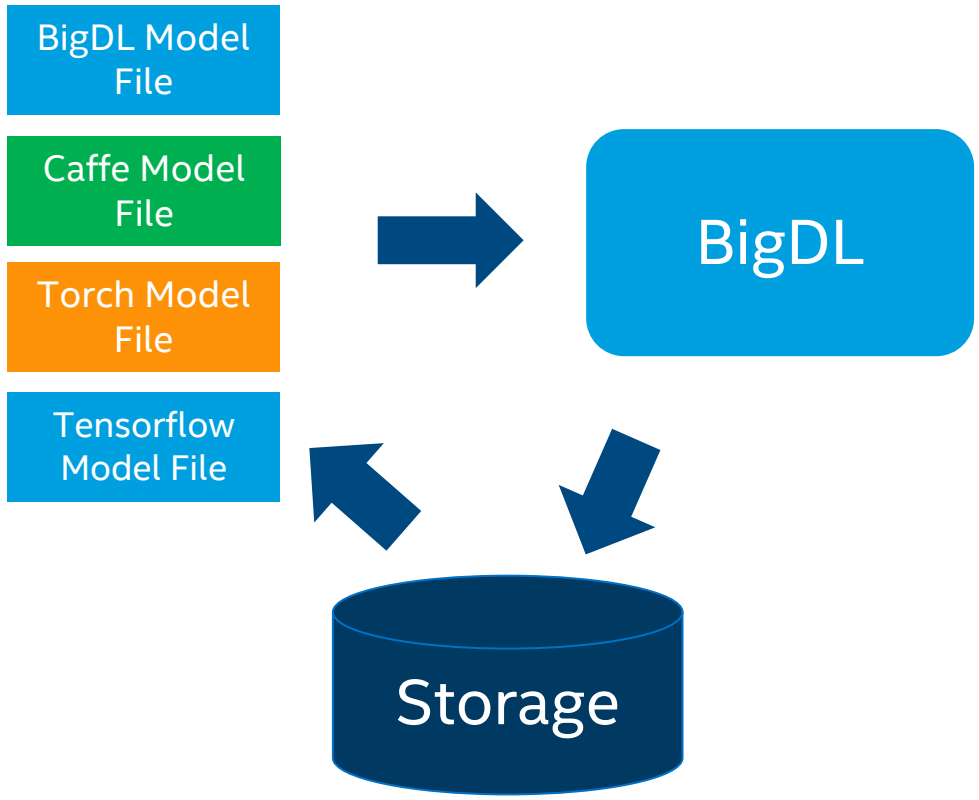
BIGDL AS A STANDARD SPARK PROGRAM

- Distributed Deep learning applications (training, fine-tuning & prediction) on Apache Spark*
  - No changes to the existing Hadoop/Spark clusters needed



MODELS INTEROPERABILITY SUPPORT

- Model Snapshot
  - Long training work checkpoint
  - Model deployment and sharing
  - Fine-tune
- Caffe/Torch/Tensorflow Model Support
  - Model file load
  - Easy to migrate your Caffe/Torch/Tensorflow work to Spark
- **NEW** - BigDL supports loading pre-defined Keras models (Keras 1.2.2)



BIGDL: PYTHON API

- Support deep learning model training, evaluation, inference
- Support Spark v1.5/1.6/2.X
- Support **Python 2.7/3.5/3.6**
- Based on PySpark, **Python API** in BigDL allows use of existing Python libs (Numpy, Scipy, Pandas, Scikit-learn, NLTK, Matplotlib, etc)

```
train_data = get_minst("train").map(
    normalizer(mnist.TRAIN_MEAN, mnist.TRAIN_STD))
test_data = get_minst("test").map(
    normalizer(mnist.TEST_MEAN, mnist.TEST_STD))
state = {"batchSize": int(options.batchSize),
         "learningRate": 0.01,
         "learningRateDecay": 0.0002}
optimizer = Optimizer(
    model=build_model(10),
    training_rdd=train_data,
    criterion=ClassNLLCriterion(),
    optim_method="SGD",
    state=state,
    end_trigger=MaxEpoch(100))
optimizer.setvalidation(
    batch_size=32,
    val_rdd=test_data,
    trigger=EveryEpoch(),
    val_method=["top1"]
)
optimizer.setcheckpoint(EveryEpoch(), "/tmp/lenet5/")
trained_model = optimizer.optimize()
```

WORKS WITH NOTEBOOK

- Running BigDL applications directly in Jupyter, Zeppelin, Databricks notebooks, etc.

✓Share and Reproduce

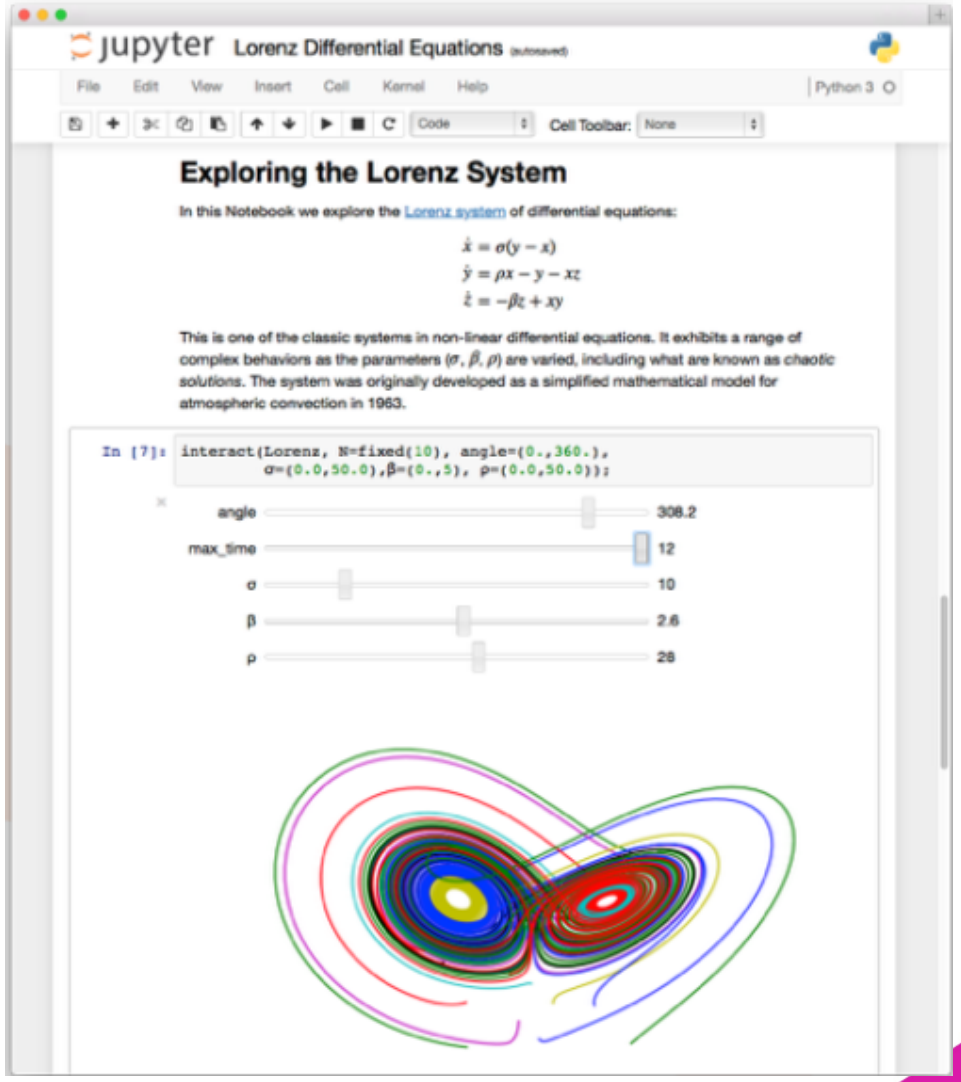
- Notebooks can be shared with others
- Easy to reproduce and track

✓Rich Content

- Texts, images, videos, LaTeX and JavaScript
- Code can also produce rich contents

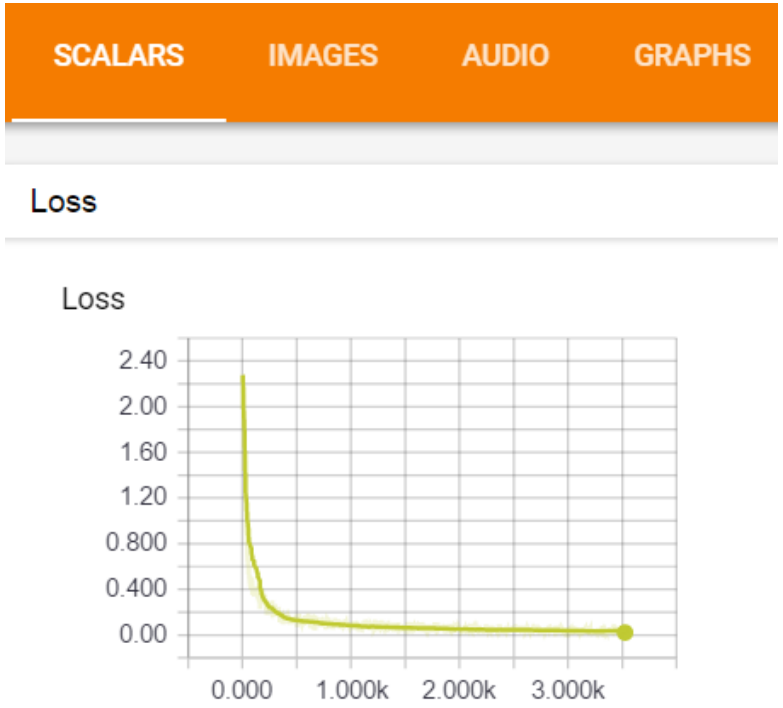
✓Rich toolbox

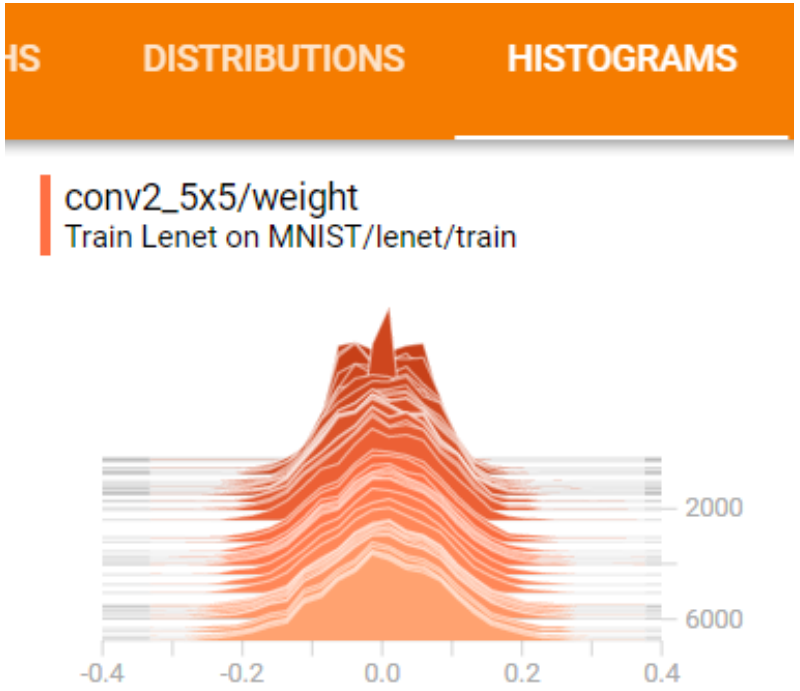
- Apache Spark, from Python, R and Scala
- Pandas, scikit-learn, ggplot2, dplyr, etc



VISUALIZATION FOR LEARNING

- BigDL integration with TensorBoard
  - TensorBoard is a suite of web applications from Google for visualizing and understanding deep learning applications





CURRENT RELEASE BIGDL 0.5.0

- Support more Tensorflow operations, e.g., loading Tensorflow dynamic models (e.g. LSTM, RNN) in BigDL
- Support combining data pre-processing and neural network layers in the same model (to make model deployment easy)
- Keras-like APIs (Scala and Python) for users to run their Keras code on BigDL
- Speedup various modules in BigDL (BCECriterion, RMSprop, LeakyRelu, etc.)
- Add DataFrame-based image reader and transformer

Please refer to the release note at https://github.com/intel-analytics/BigDL/releases/tag/v0.5.0 for more details

BIGDL ANALYTICS ZOO

Analytics + AI Pipelines for Spark and BigDL

“Out-of-the-box” ready for use

- Reference use cases
  - Fraud detection, time series prediction, sentiment analysis, chatbot, etc.
- Predefined models
  - Object detection, image classification, text classification, recommendations, etc.
- Feature transformations
  - Vision, text, 3D imaging, etc.
- High level APIs
  - DataFrames, ML Pipelines, Keras/Keras2, etc.

BUILDING AND DEPLOYING WITH BIGDL

TECHNOLOGY

bluedata
cloudera
CRAY THE SUPERCOMPUTER COMPANY
databricks
DELL EMC
KINGSOFT
Lightbend
Qubole

CLOUD SERVICE PROVIDERS

Alibaba Cloud aliyun.com
aws
Cloud Dataproc
Azure
IBM Cloud

END USERS

cdhi
GIGASPACES innovate with confidence
JD.COM 京东
UnionPay 银联

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

BIGDL WORKLOADS…..ACROSS THE INDUSTRY

CONSUMER

CALL CENTER ROUTING
IMAGE SIMILARITY SEARCH
SMART JOB SEARCH

HEALTH

ANALYSIS OF 3D MRI MODELS FOR KNEE DEGRADATION

FINANCE

FRAUD DETECTION
RECOMMENDATION
CUSTOMER/MERCHANT PROPENSITY

RETAIL

IMAGE FEATURE EXTRACTION

MANUFACTURING

STEEL SURFACE DEFECT DETECTION

SCIENTIFIC COMPUTING

WEATHER FORECASTING

AND OTHER EMERGING USAGES…

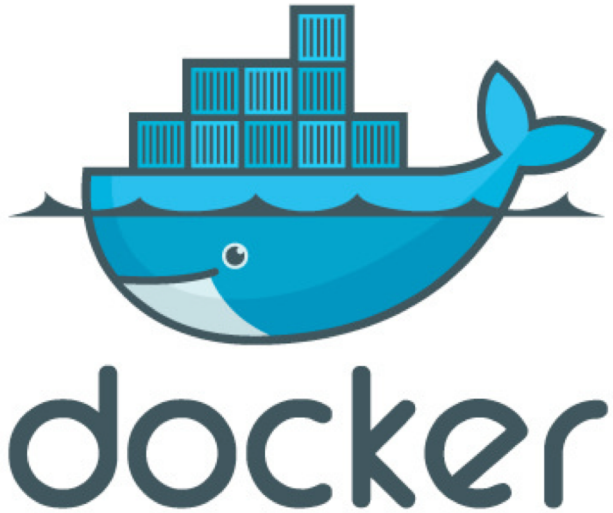
BIGDL HANDS ON LAB SETUP

RUNNING BIGDL

- Step 1: Setup env
  - On your laptop
    - Option 1 : Docker
    - Option 2 : ElephantScale Sandbox (Virtual machine)
  - In the cloud
    - Option 3 : AWS : AMI
- Step 2 : Get sample tutorials

HANDS ON LAB SETUP

OPTION 1 - DOCKER



To do todays lab using Docker you will need to use the following steps:

- Please install Docker http://docker.com on your local machine
- Go to https://github.com/intel-analytics/BigDL-trainings
- Clone repo

HANDS ON LAB SETUP

OPTION 2 – VIRTUAL MACHINE

Lab materials your will need to use a local VM instance:

- Please download OVA file http://elephantscale.com/sandbox
- Install VMWare or VirtualBox

HANDS ON LAB SETUP

OPTION 3 – AWS VIRTUAL MACHINE

- Go to https://AWS.com
- Install AWS credits
- Spin up the following AMI
  - TODO : provide AMI ID (can be in the labs)

BIGDL HANDS ON LAB PROJECTS

RUNNING BIGDL

GETTING TUTORIALS

Lab Setup:

- Clone Repo
  - Git clone https://www.github.com/intel.analytics/BigDL-trainings
- If using DOCKER
  - Run "run-bidgl-docker.sh"
- If running natively
  - Edit ***"run-bigdl-native.sh"*** script for your env
  - Run ***"run-bigdl-native"***
- Open Jupyter notebook

HANDS ON LAB - PROJECT 1

CREATING A BASIC NEURAL NETWORK

- Create Feed-Forward-Network Example
- IRIS Classifier Example
  - We will show how to run a basic classifier using the classic Iris dataset
- Credit Card Fraud Detection Example
  - We will classify credit card transactions as fraud / legitimate

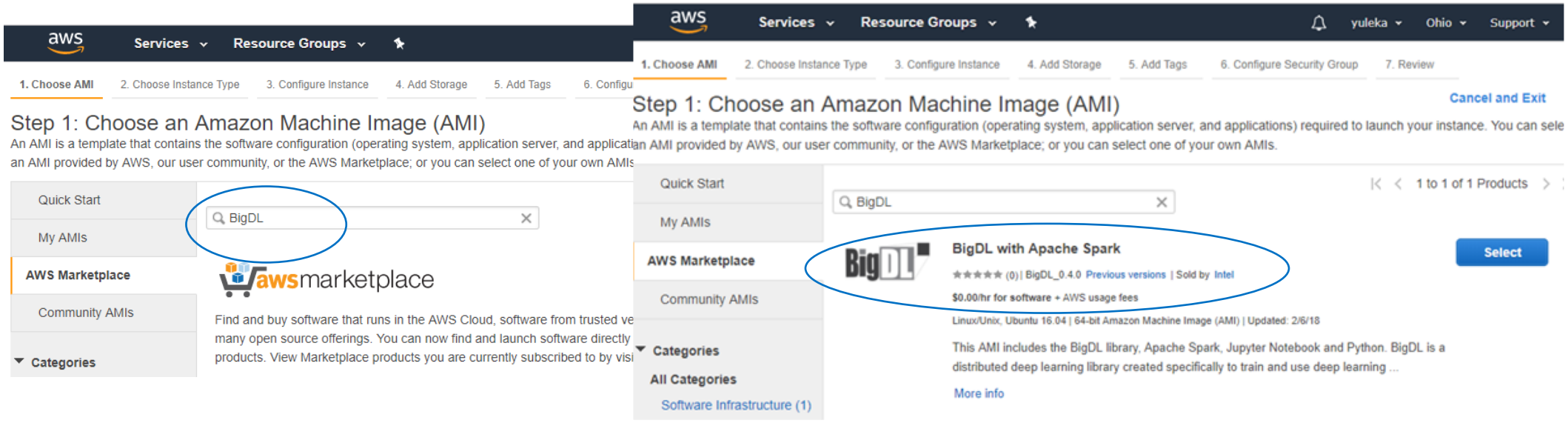
HANDS ON - PROJECT 2

CREATING A LONG SHORT TERM MEMORY (LSTM) NEURAL NETWORK

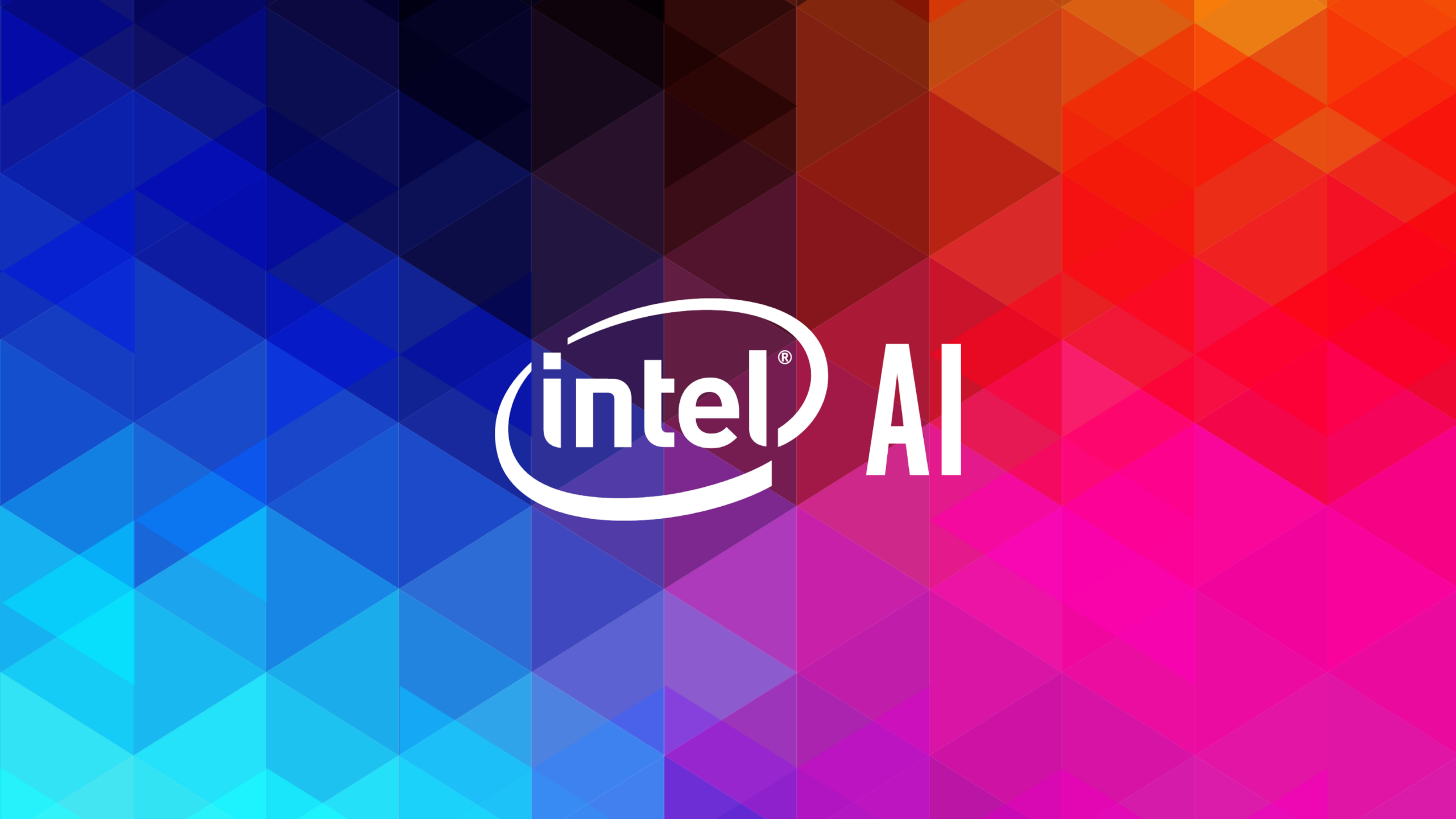
- Analysis of Newsgroup Data Example
  - We will use newsgroup postings to classify posts

CALL TO ACTION

- Build with BigDL on AWS - lookup BigDL AMI on AWS Marketplace



- Find more information on BigDL at https://github.com/intel-analytics/BigDL


intel®
AI