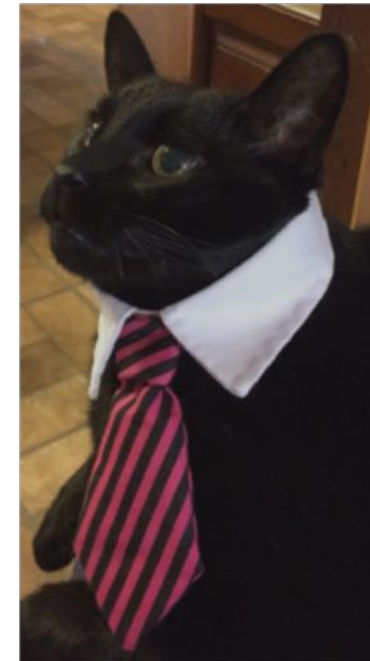


Apache Deep Learning 101 v0.31

(For Data Engineers)



```
{  
  "prediction": [  
    {  
      "class": "n02825657 bell cote, bell cot",  
      "probability": 0.49351149797439575  
    },  
    {  
      "class": "n04366367 suspension bridge",  
      "probability": 0.17974209785461426  
    },  
    {  
      "class": "n03028079 church, church building",  
      "probability": 0.11694391071796417  
    },  
    {  
      "class": "n03032252 cinema, movie theater, movie theatre, movie house, picture palace",  
      "probability": 0.07838434725999832  
    },  
    {  
      "class": "n03781244 monastery",  
      "probability": 0.04639515280723572  
    }  
  ]  
}
```



Timothy Spann
@PaaSDev

<https://github.com/tspannhw/apache-deep-learning-101/releases/tag/3.1>

Disclaimer

- This is my personal integration and use of Apache software, no companies vision.
- This document may contain product features and technology directions that are under development, may be under development in the future or may ultimately not be developed. This is Tim's ideas only.
- Technical feasibility, market demand, user feedback, and the Apache Software Foundation community development process can all effect timing and final delivery.
- This document's description of these features and technology directions does not represent a contractual commitment, promise or obligation from Hortonworks to deliver these features in any generally available product.
- Product features and technology directions are subject to change, and must not be included in contracts, purchase orders, or sales agreements of any kind.
- Since this document contains an outline of general product development plans, customers should not rely upon it when making a purchase decision.

Agenda - Data Engineering With Apache Deep Learning

- Introduction – This is my personal workflow
- Architecture Overview
- Apache NiFi 1.7
- Apache MXNet 1.3
- Apache OpenNLP and Apache Tika
- Demos
- Questions

There are some who call him..

DZone Big Data MVB

Princeton Future of Data Meetup

Ex-Pivotal Senior Field Engineer

Current Hortonworks Senior Solutions Engineer

<https://github.com/tspannhw>



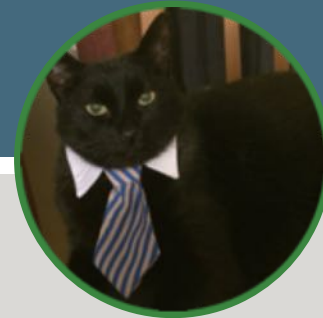
Deep Learning for Big Data Engineers

Multiple users, frameworks, languages, devices, data sources & clusters



BIG DATA ENGINEER

- Experience in ETL
- Coding skills in Scala, Python, Java
- Experience with Apache Hadoop
- Knowledge of database query languages such as SQL
- Knowledge of Hadoop tools such as Hive, or Pig



CAT

- Expert in ETL (Eating, Ties and Laziness)
- Social Media Maven
- Deep SME in Buzzwords
- No Coding Skills
- Interest in Pig and Falcon

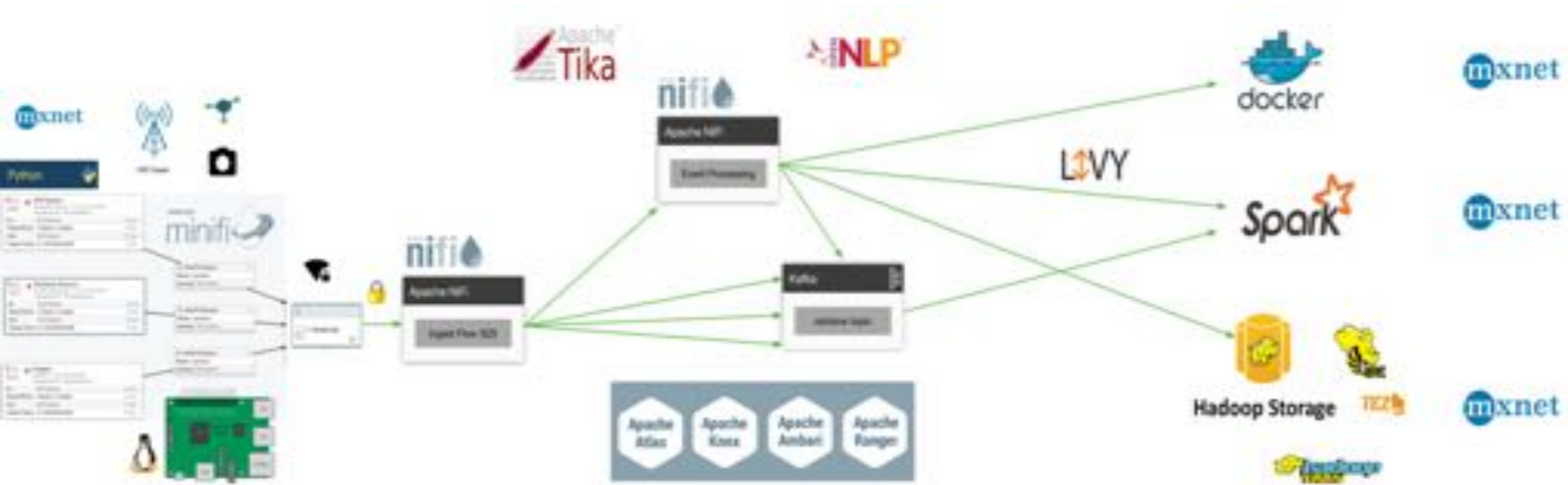


AI

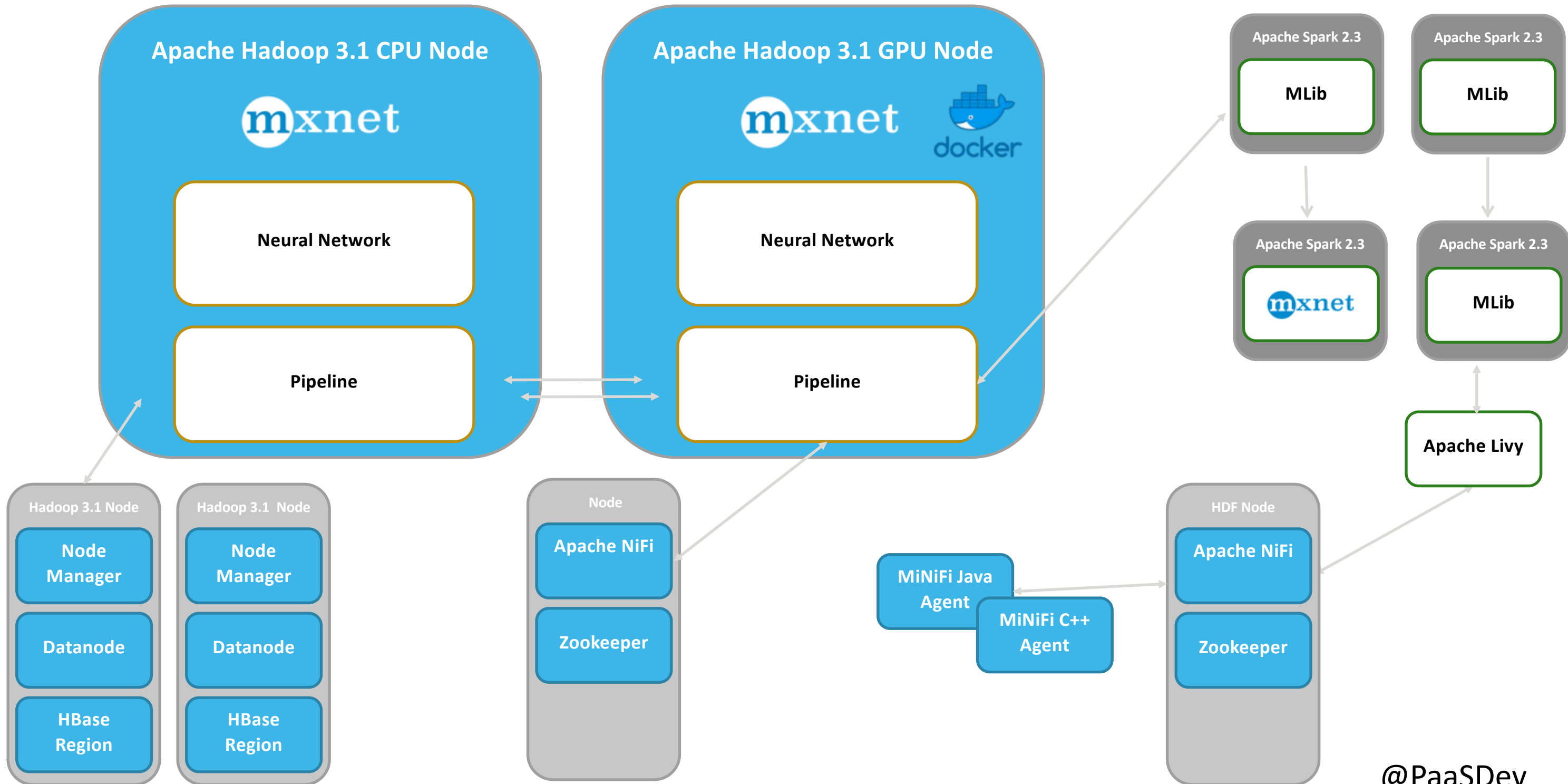
- Will Drive your Car
- Will Fix Your Code
- Will Beat You At Q-Bert
- Will Not Be Discussed Today
- Will Not Finish This Talk For Me, This Time



Apache Deep Learning Flow



Apache Deep Learning Server Architecture



Collect: Bring Together



Aggregate all the Data!
sensors, drones, logs,
geo-location devices
images from cameras,
results from running predictions on
pre-trained models.



Conduct: Mediate the Data Flow



Mediate point-to-point and bi-directional data flows delivering data reliably to and from Apache HBase, Apache Hive, HDFS, Slack and Email.

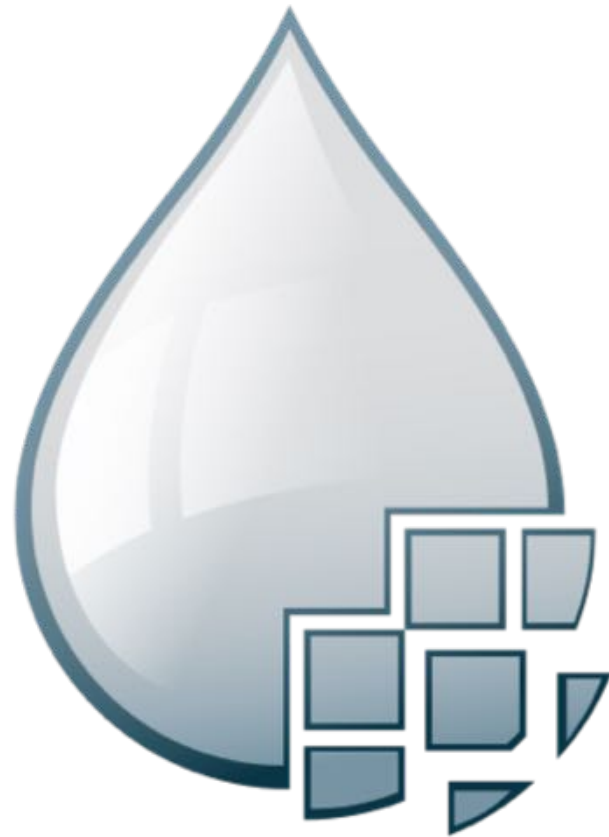


Curate: Gain Insights



Orchestrate, parse, merge, aggregate, filter, join, transform, fork, query, sort, dissect, store, enrich with weather, location, sentiment analysis, image analysis, object detection, image recognition

Why Apache NiFi?



- Guaranteed delivery
- Data buffering
 - Backpressure
 - Pressure release
- Prioritized queuing
- Flow specific QoS
 - Latency vs. throughput
 - Loss tolerance
- Data provenance
- Supports push and pull models
- Hundreds of processors
- Visual command and control
- Over a sixty sources
- Flow templates
- Pluggable/multi-role security
- Designed for extension
- Clustering
- Version Control

Edge Intelligence with Apache NiFi Subproject - MiNiFi

Key Features

- ◆ Guaranteed delivery
- ◆ Data buffering
 - Backpressure
 - Pressure release
- ◆ Prioritized queuing
- ◆ Flow specific QoS
 - Latency vs. throughput
 - Loss tolerance
- ◆ Data provenance



- ◆ Recovery / recording a rolling log of fine-grained history
- ◆ Designed for extension

Different from Apache NiFi

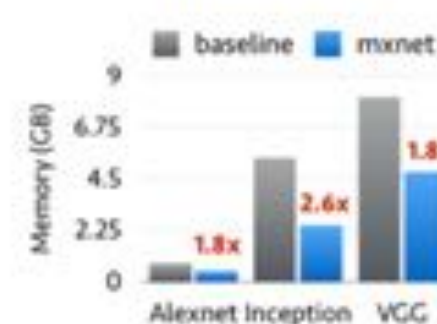
- ◆ Design and Deploy
- ◆ Warm re-deploys



Portable



Efficient



Scalable



- Cloud ready
- Python, C++, Scala, R, Julia, Matlab, MXNet.js and Perl Support
- Experienced team (**XGBoost**)
- AWS, Microsoft, NVIDIA, Baidu, Intel
- Apache Incubator Project
- Run distributed on YARN and Spark
- In my early tests, faster than TensorFlow. (Try this your self)
- Runs on Raspberry PI, NVidia Jetson TX1 and other constrained devices

<https://github.com/apache/incubator-mxnet/tree/1.3.0/example>

https://mxnet.incubator.apache.org/how_to/cloud.html

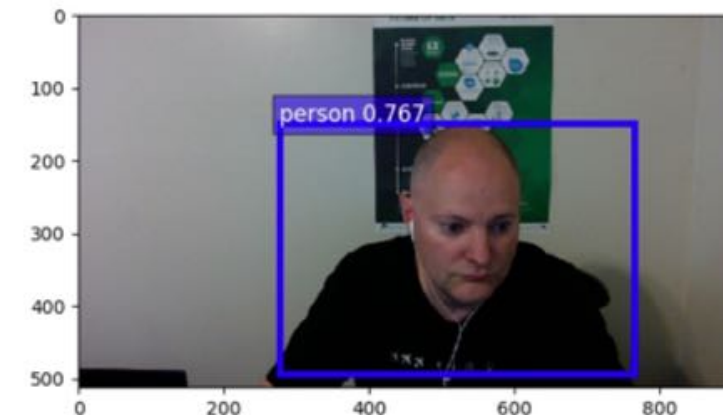
https://gluon-cv.mxnet.io/api/model_zoo.html



- Great documentation
- Crash Course
- **Gluon (Open API), GluonCV, GluonNLP**
- **Keras (One API Many Runtime Options)**
- Great Python Interaction
- Model Server Available
- **ONNX (Open Neural Network Exchange Format) Support for AI Models**
- Now in Version 1.3
- Rich Model Zoo!
- TensorBoard compatible

<https://onnx.ai/> <http://mxnet.incubator.apache.org/> <http://gluon.mxnet.io/> <https://gluon-nlp.mxnet.io/>

pip3.6 install -U keras-mxnet pip3.6 install --pre --upgrade mxnet pip3.6 install gluonnlp



Apache NiFi Integration with Apache MXNet Options

- Apache MXNet via Execute Process (Python)

<https://community.hortonworks.com/articles/198939/using-apache-mxnet-gluoncv-with-apache-nifi-for-de.html>

- Apache MXNet Running on Edge Nodes (MiniFi) S2S

<https://community.hortonworks.com/articles/176932/apache-deep-learning-101-using-apache-mxnet-on-the.html>

- Apache MXNet Model Server Integration (REST API)

<https://community.hortonworks.com/articles/177232/apache-deep-learning-101-processing-apache-mxnet-m.html>

Apache NiFi Integration with Apache Hadoop Options

- Apache MXNet Running in Apache Zeppelin Notebooks

<https://community.hortonworks.com/articles/176789/apache-deep-learning-101-using-apache-mxnet-in-apa.html>

- Apache MXNet Running on YARN 3.1 In Hadoop 3.1 In Dockerized Containers

https://www.slideshare.net/Hadoop_Summit/deep-learning-on-yarn-running-distributed-tensorflow-etc-on-hadoop-cluster-v3

- Apache MXNet Running on YARN

<https://community.hortonworks.com/articles/174399/apache-deep-learning-101-using-apache-mxnet-on-apa.html>

Apache MXNet Pre-Built Models - Model Zoo

- CaffeNet
- SqueezeNet v1.1
- Inception v3
- Single Shot Detection (SSD)
- VGG16
- VGG19
- ResidualNet 152
- LSTM



http://mxnet.incubator.apache.org/model_zoo/index.html

http://mxnet.incubator.apache.org/api/python/gluon/model_zoo.html

Installing Apache MXNet on a Raspberry Pi

<https://mxnet.incubator.apache.org/install/index.html?platform=Devices&language=Python&processor=CPU>

I highly recommend using Python 3.6 and full custom build of OpenCV 3.x.

See: https://mxnet.incubator.apache.org/tutorials/embedded/wine_detector.html

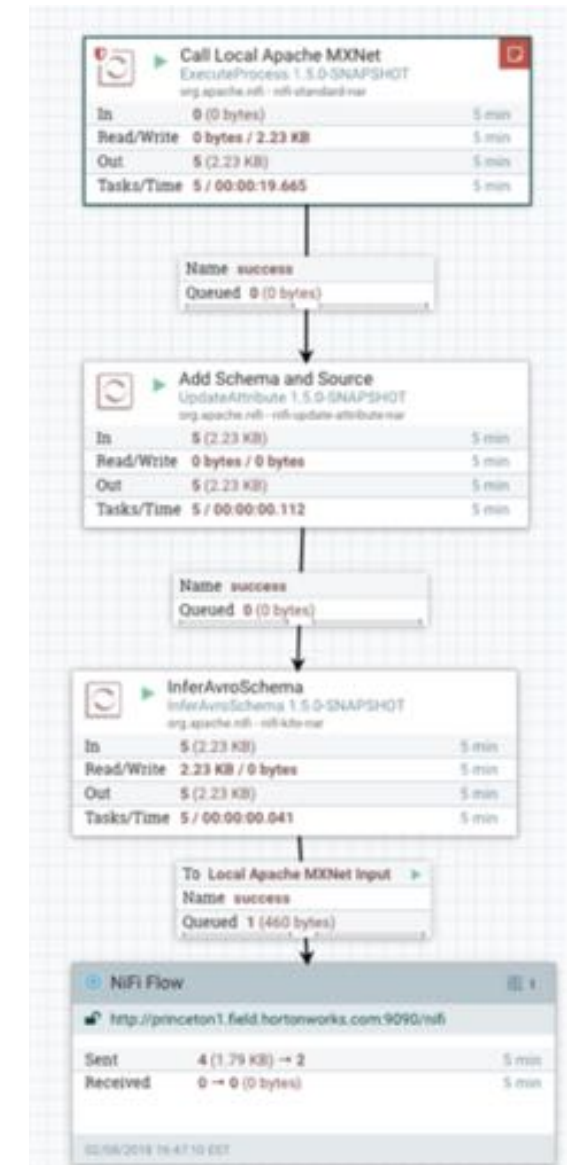


Apache MXNet via Python (OSX Local with WebCam)

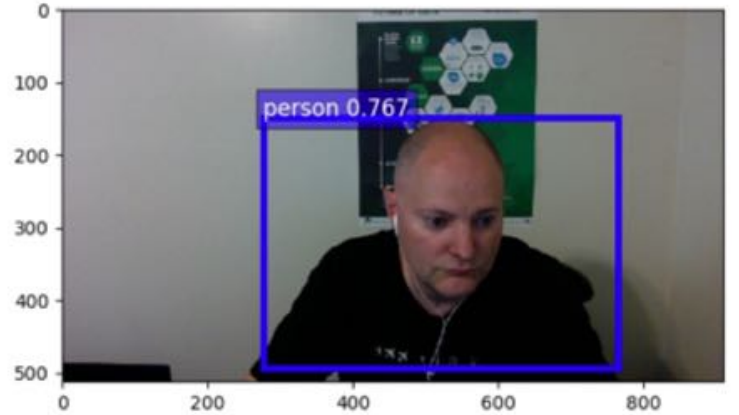
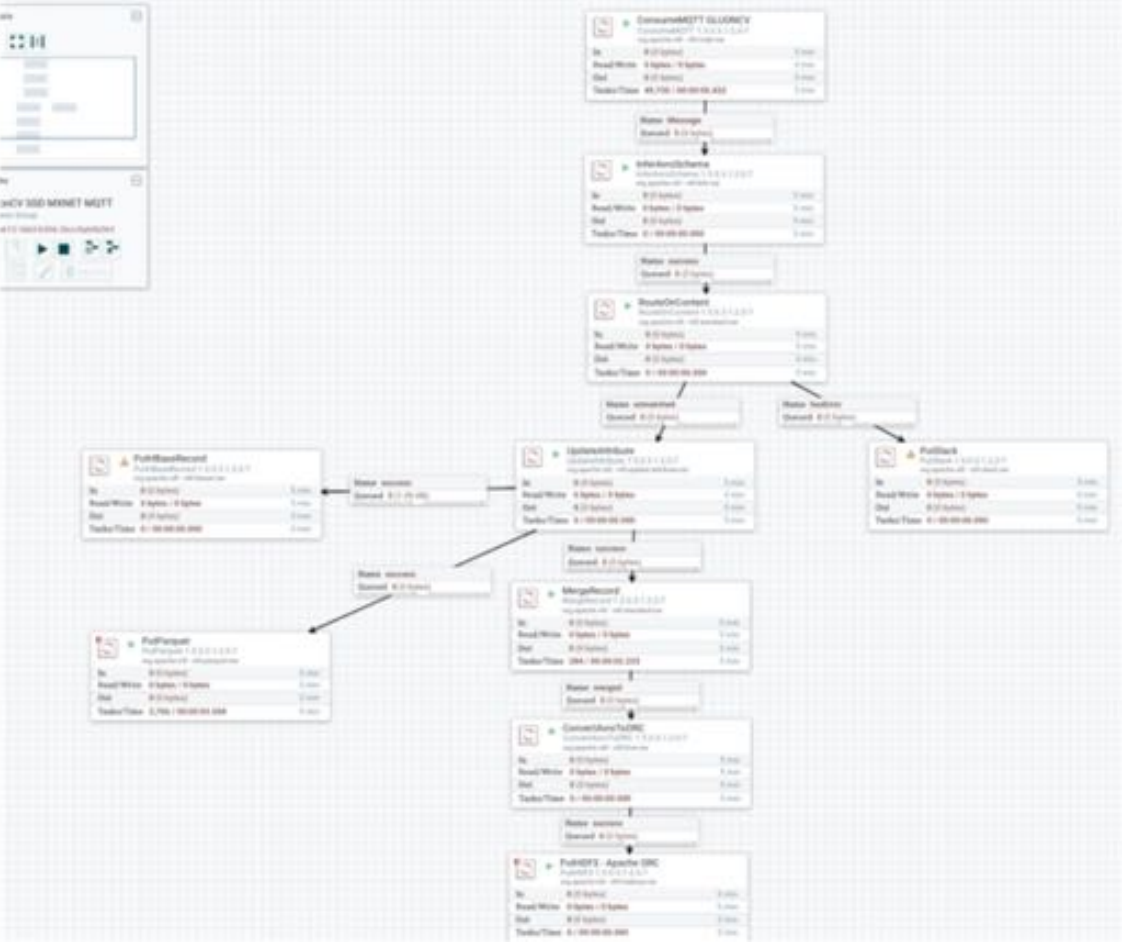
```
python3 -W ignore analyze.py
```

```
{"uuid": "mxnet_uuid_img_20180208204131", "top1pct": "30.0999999046", "top1":  
"n02871525 bookshop, bookstore, bookstall", "top2pct": "23.7000003457", "top2":  
"n04200800 shoe shop, shoe-shop, shoe store", "top3pct": "4.80000004172", "top3":  
"n03141823 crutch", "top4pct": "2.89999991655", "top4": "n04370456 sweatshirt",  
"top5pct": "2.80000008643", "top5": "n02834397 bib", "imagefilename":  
"images/tx1_image_img_20180208204131.jpg", "runtime": "2"}
```

<https://community.hortonworks.com/articles/171960/using-apache-mxnet-on-an-apache-nifi-15-instance-w.html>



Apache MXNet with Gluon and MQTT



label	confidence	label	confidence	label	confidence	label	confidence	label	confidence	label	confidence
cellular telephone, cellular phone, cellphone, cell, mobile phone	48.4	rubber eraser, rubber, pencil eraser	14.3	remote control, remote	9.1	remote control, remote	8.2	remote control, remote	4.7	remote control, remote	4.7
tebbi, tebbi bear	76.6	remote control, remote	3.9	remote control, remote	3.2	remote control, remote	3.2	remote control, remote	2.9	remote control, remote	2.9
macaw	7.8	remote control, remote	6.8	remote control, remote	5.7	remote control, remote	5.4	remote control, remote	3.0	remote control, remote	3.0
IPad	12.5	remote control, remote	10.9	remote control, remote	8.5	remote control, remote	8.2	remote control, remote	4.7	remote control, remote	4.7
cellphone	16.8	remote control, remote	10.1	remote control, remote	7.3	remote control, remote	5.4	remote control, remote	5.3	remote control, remote	5.3
nematode, nematode worm, roundworm	8.7	remote control, remote	8.2	remote control, remote	6.9	remote control, remote	4.8	remote control, remote	4.2	remote control, remote	4.2
lender, ring binder	7.8	remote control, remote	4.3	remote control, remote	4.0	remote control, remote	3.4	remote control, remote	3.2	remote control, remote	3.2

<https://community.hortonworks.com/articles/198939/using-apache-mxnet-gluoncv-with-apache-nifi-for-de.html>

<https://community.hortonworks.com/articles/198912/ingesting-apache-mxnet-gluon-deep-learning-results.html>

Apache MXNet 1.3 with GluonCV YOLO v3 and Apache NiFi



<https://community.hortonworks.com/articles/222367/using-apache-nifi-with-apache-mxnet-gluoncv-for-yo.html>

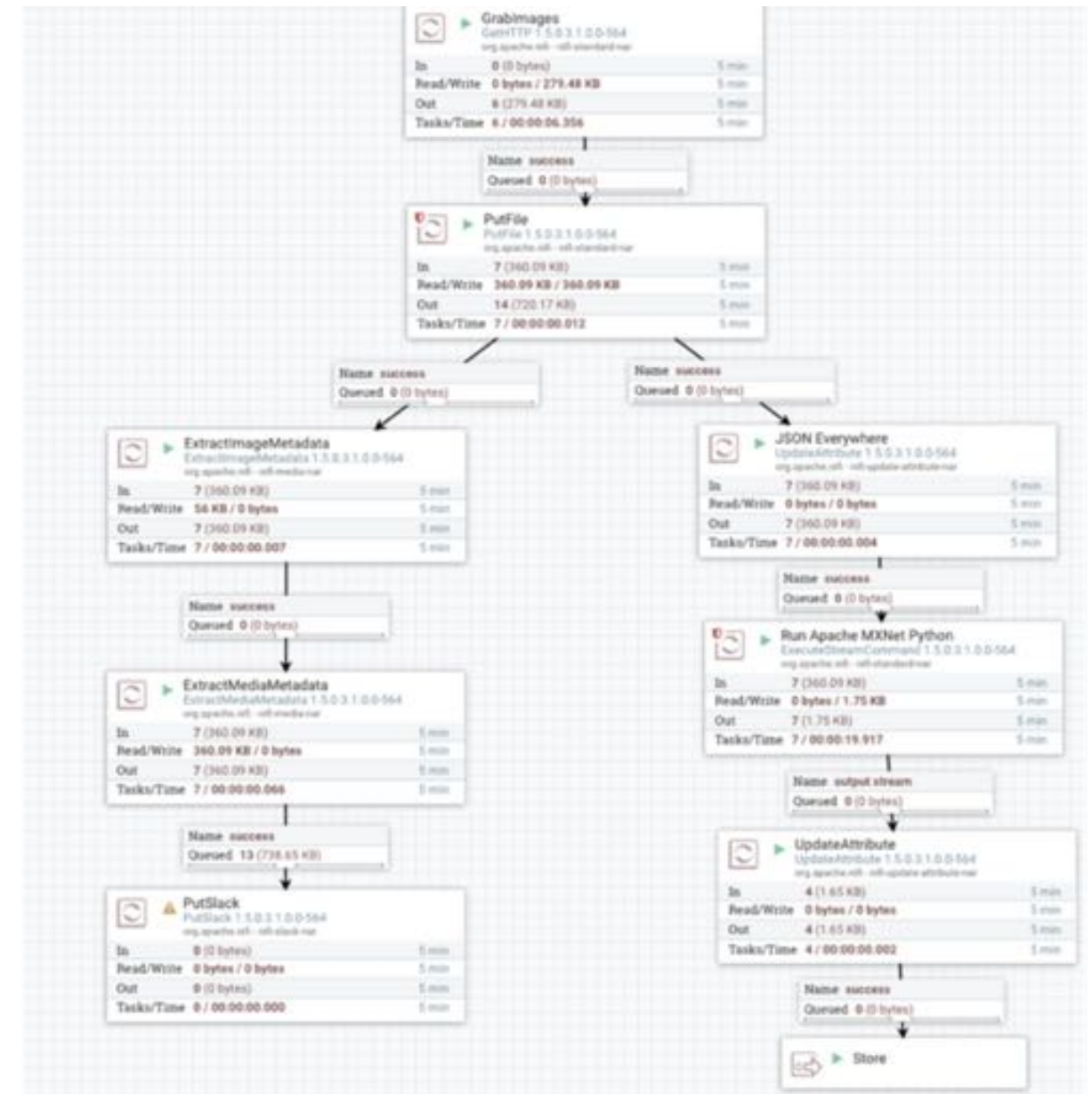
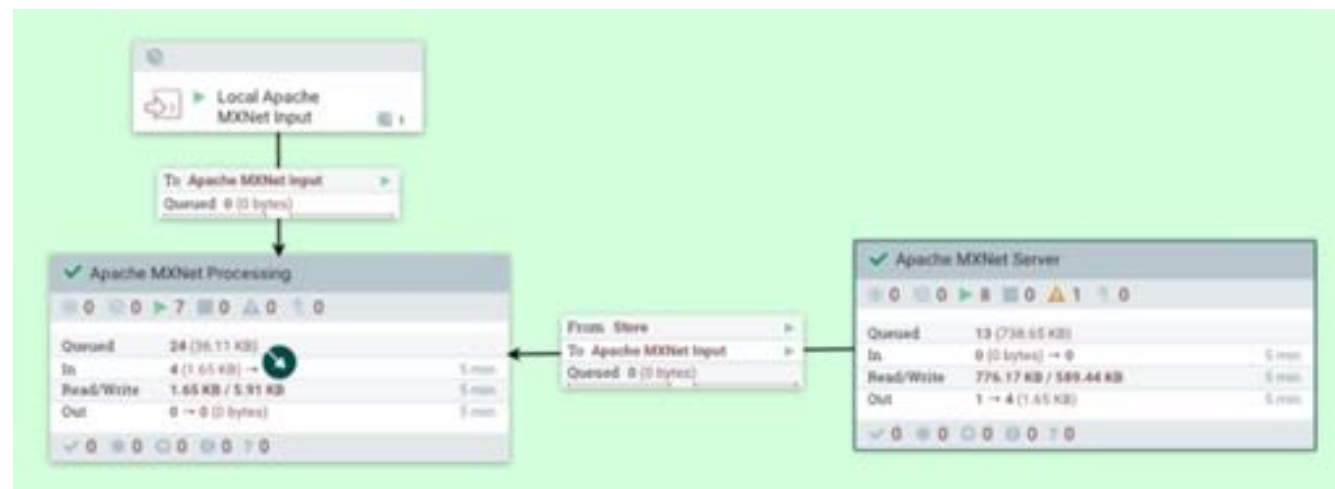
Apache MXNet Installation on OSX

```
git clone https://github.com/apache/incubator-mxnet.git
cd incubator-mxnet
mkdir images
curl --header 'Host: data.mxnet.io' --header 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:45.0) Gecko/20100101 Firefox/45.0' --header 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8' --header 'Accept-Language: en-US,en;q=0.5' --header 'Referer: http://data.mxnet.io/models/imagenet/' --header 'Connection: keep-alive' 'http://data.mxnet.io/models/imagenet/inception-bn.tar.gz' -o 'inception-bn.tar.gz' -L
tar -xvzf inception-bn.tar.gz
cp Inception-BN-0126.params Inception-BN-0000.params
brew install graphviz
pip install --upgrade pip
pip install --upgrade setuptools
pip install graphviz
pip install mxnet
```

<http://mxnet.incubator.apache.org/install/index.html>

Apache MXNet Running on an Apache NiFi Node

Property	Value
Command Arguments	"/opt/demo/images/\${filename}"
Command Path	/opt/demo/incubator-mxnet/centosrun.sh
Ignore STDIN	true
Working Directory	/opt/demo/incubator-mxnet/
Argument Delimiter	;
Output Destination Attribute	No value set
Max Attribute Length	51920



Apache MXNet Installation on an Centos 7

```
git clone https://github.com/apache/incubator-mxnet.git
```

```
sudo yum groupinstall 'Development Tools' -y
```

```
sudo yum install cmake git pkgconfig -y
```

```
sudo yum install libpng-devel libjpeg-turbo-devel jasper-devel openexr-devel  
libtiff-devel libwebp-devel -y
```

```
sudo yum install libdc1394-devel libv4l-devel gstreamer-plugins-base-devel -y
```

```
sudo yum install gtk2-devel -y
```

```
sudo yum install tbb-devel eigen3-devel -y
```

```
pip install numpy
```

You will need a full Python development environment, C++ and I recommend building OpenCV3.

<https://community.hortonworks.com/articles/174227/apache-deep-learning-101-using-apache-mxnet-on-an.html>

Apache MXNet Running on Edge Nodes (MiniFi)



```
063443271 goblet
063242569 drum, membranophone, tympan
044823962 punching bag, punch bag, punching ball, poollah
04447061 toilet seat
(0, 1)
Capturing
Predicting
pre-processed image is 0.170602932795
forward pass in 4.7605560182
probability=0.309425, class=063637318 lampshade, lamp shade
probability=0.116924, class=063443271 goblet
probability=0.095156, class=04306533 table lamp
probability=0.076046, class=04823962 punching bag, punch bag, punching ball, poollah
probability=0.037213, class=04447061 toilet seat
((0.30341567, '063637318 lampshade, lamp shade'), (0.1169137, '063443271 goblet'), (0.095149919, '04306533 table lamp'), (0.076046, '04823962 punching bag, punch bag, punching ball, poollah'), (0.037213, '04447061 toilet seat'))
let seat')
063637318 lampshade, lamp shade
063443271 goblet
04306533 table lamp
04823962 punching bag, punch bag, punching ball, poollah
04447061 toilet seat
(0, 1)
Capturing
Predicting
pre-processed image is 0.170304083881
forward pass in 4.07004500127
probability=0.243294, class=063637318 lampshade, lamp shade
probability=0.209549, class=04823962 punching bag, punch bag, punching ball, poollah
probability=0.104245, class=04447061 toilet seat
probability=0.067596, class=04306533 table lamp
probability=0.030563, class=063242569 drum, membranophone, tympan
((0.24329425, '063637318 lampshade, lamp shade'), (0.20954977, '04823962 punching bag, punch bag, punching ball, poollah'), (0.104245, '063637318 lampshade, lamp shade'), (0.067596, '04823962 punching bag, punch bag, punching ball, poollah'), (0.030563, '04306533 table lamp'), (0.030563, '063242569 drum, membranophone, tympan'))
(0, 1)
Capturing
Predicting
pre-processed image is 0.100067062370
Traceback (most recent call last):
  File "mqttinfi2.py", line 44, in __main__
    img = inception_predict_product_from_local_file(filename, 0)
  File "/usr/lib/python3.7/site-packages/inception_predict.py", line 72, in predict_from_local_file
    return predict(filename, sub_image, 0)
  File "/usr/lib/python3.7/site-packages/inception_predict.py", line 47, in predict
```



<https://github.com/tspannhw/OpenSourceComputerVision>

<https://github.com/tspannhw/ApacheDeepLearning101>

<https://github.com/tspannhw/mxnet-for-iot>

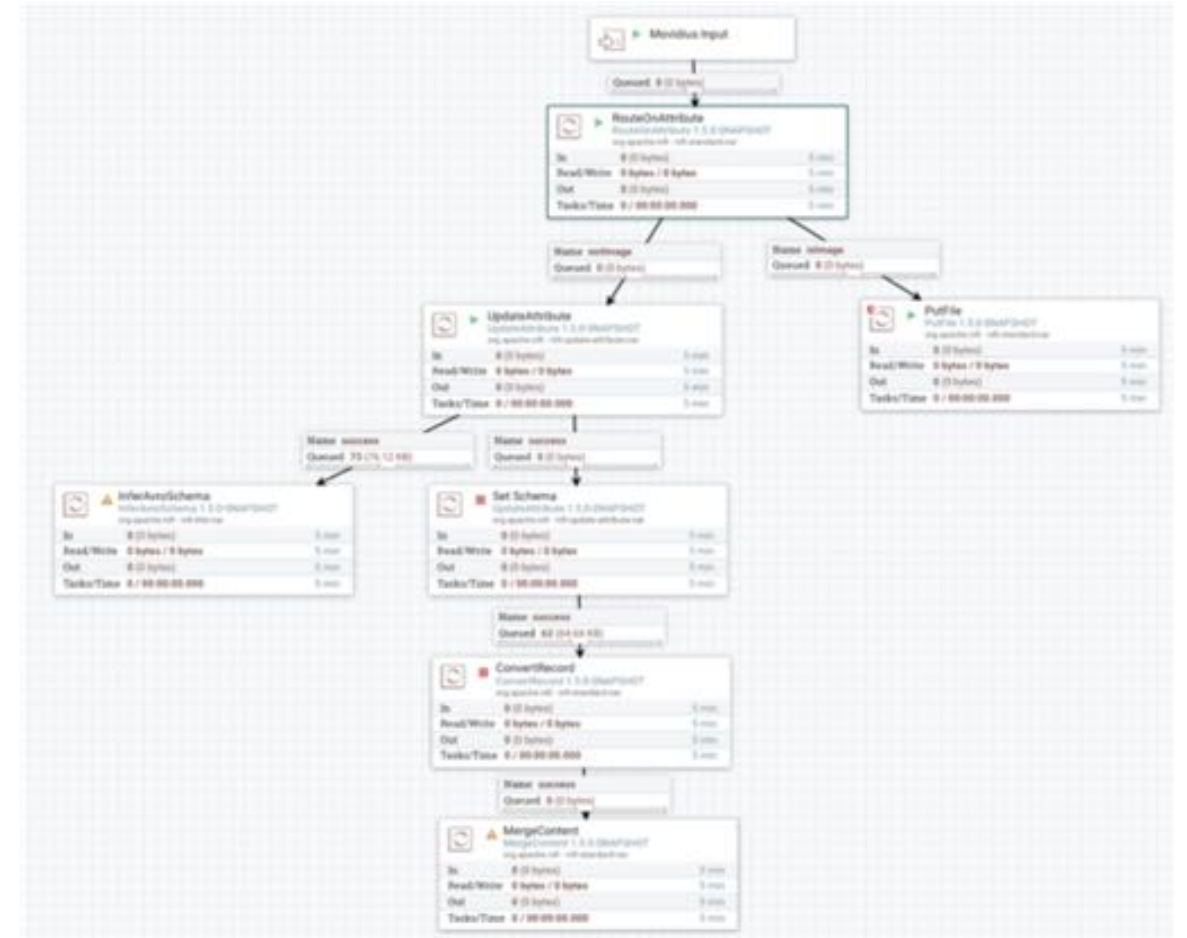
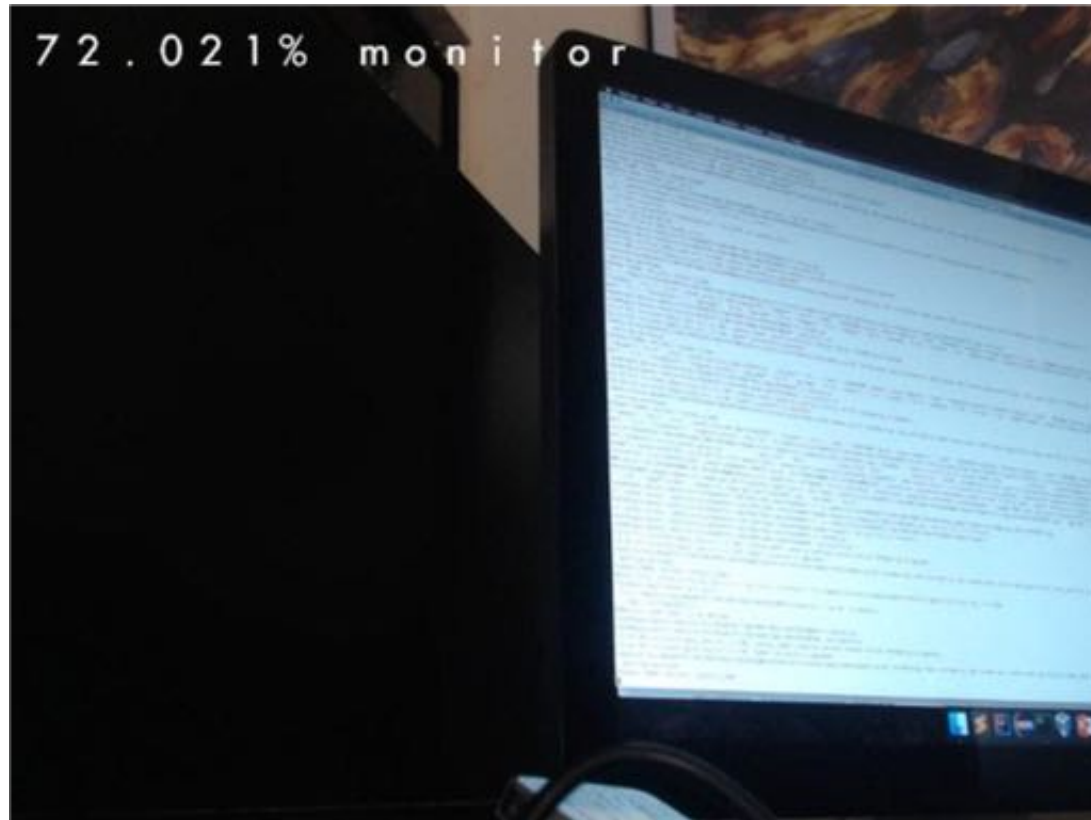
<https://community.hortonworks.com/articles/83100/deep-learning-iot-workflows-with-raspberry-pi-mqtt.html>

Multiple IoT Devices with Apache NiFi and Apache MXNet

The screenshot displays the Apache NiFi web console. At the top, a SQL query is entered: `select latitude, longitude, is from gps`. Below the query, there is a toolbar with various icons for map interaction. The 'Available Fields' section shows four fields: 'latitude', 'longitude', 'is', and 'gps'. Each field has a corresponding input box with a dropdown arrow. Below the fields, a map visualization is shown, featuring a road network with red and orange lines, green areas representing parks or forests, and a blue pin marker. The map includes a zoom control in the top-left corner and a copyright notice 'Lafel | © OpenStreetMap contributors' in the bottom-right corner.

<https://community.hortonworks.com/articles/203638/ingesting-multiple-iot-devices-with-apache-nifi-17.html>

Using Apache MXNet on The Edge with Sensors and Intel Movidius (MiniFi)



<https://community.hortonworks.com/articles/146704/edge-analytics-with-nvidia-jetson-tx1-running-apac.html>

<https://community.hortonworks.com/articles/176932/apache-deep-learning-101-using-apache-mxnet-on-the.html>

Apache MXNet Setup in Apache Zeppelin

Deep Learning Models

You will need to download the pre-built Inception models and reference them on your server.

synset.txt

Inception-BN-0000.params

Inception-BN-symbol.json

See: https://mxnet.incubator.apache.org/tutorials/embedded/wine_detector.html

curl <http://data.mxnet.io/models/imagenet/inception-bn.tar.gz> > inception-bn.tar.gz

curl <http://data.mxnet.io/models/imagenet/synset.txt> > synset.txt



```
mxnet Notebook - Job
mxnet
ipython

import time
import sys
import datetime
import subprocess
import sys
import os
import datetime
import traceback
import math
import random, string
import base64
import json
from time import gettime, strftime
import mxnet as mx
import numpy as np
import math
import random, string
import time
from time import gettime, strftime
# Forked from Apache MXNet example with minor changes for osx
import time
import mxnet as mx
import numpy as np
import cv2, os, urllib
from collections import namedtuple
Batch = namedtuple('Batch', ['data'])

# Load the symbols for the networks
with open('/opt/demo/incubator-mxnet/synset.txt', 'r') as f:
    synsets = [l.rstrip() for l in f]

# Load the network parameters
sym, arg_params, aux_params = mx.model.load_checkpoint('/opt/demo/incubator-mxnet/Inception-BN', 0)

# Load the network into an MXNet module and bind the corresponding parameters
mod = mx.mod.Module(symbol=sym, context=mx.gpu())
mod.bind(for_training=False, data_shapes=[('data', (1,3,224,224))])
mod.set_params(arg_params, aux_params)

...

function to predict objects by giving the model a pointer to an image file and running a forward pass through the model.

inputs:
filename - jpeg file of image to classify objects in
mod - the module object representing the loaded model
synsets - the list of symbols representing the model
N - optional parameter denoting how many predictions to return (default is top 5)

outputs:
python list of top N predicted objects and corresponding probabilities
...

def predict(filename, mod, synsets, N=5):
    tic = time.time()
    img = cv2.cvtColor(cv2.imread(filename), cv2.COLOR_BGR2RGB)
    if img is None:
```

Apache MXNet on Apache YARN 2.x Installation

```
yum install java-1.8.0-openjdk
yum install java-1.8.0-openjdk-devel
pip2.7 install kubernetes
pip2.7 install opencv-python
```

```
git clone https://github.com/dmlc/dmlc-core.git
cd dmlc-core
make
cd tracker/yarn
./build.sh
```

```
export HADOOP_HOME=/usr/hdp/3.0.0.0-1634/hadoop
export HADOOP_HDFS_HOME=/usr/hdp/3.0.0.0-1634/hadoop-hdfs
export hdfs_home=/usr/hdp/3.0.0.0-1634/hadoop-hdfs
export hadoop_hdfs_home=/usr/hdp/3.0.0.0-1634/hadoop-hdfs
```

```
git clone https://github.com/tspannhw/nifi-mxnet-yarn.git
```

```
git clone https://github.com/apache/incubator-mxnet.git
```

```
git clone https://github.com/tspannhw/ApacheDeepLearning101.git
```

Apache MXNet on Apache YARN 2.x with DMLC Script

```
dmlc-submit --cluster yarn --num-workers 1 --server-cores 2  
--server-memory 1G --log-level DEBUG --log-file mxnet.log analizyarn.py
```

The screenshot shows the Hadoop YARN web interface for an application named 'application_1517883514475_0588'. The interface includes a navigation menu on the left with options like 'Home', 'Applications', 'Jobs', 'Containers', 'Logs', and 'Metrics'. The main content area displays application details:

- Name:** dmlc-submit
- Application Type:** yarn
- Application Priority:** 0 (High) (High) (Low) (Normal) (Low) (High) (Low)
- YarnApplicationState:** FINISHED
- Queue:** default
- FinalStatus Reported by RM:** SUCCEEDED
- Started:** Sun Feb 28 08:07:40 2016
- Elapsed:** 1m
- Tracking URL:** http://10.10.10.10:8021
- Log Aggregation Status:** SUCCEEDED
- Diagnoses:** Diagnostic, Task, Submit, Transport, Heartbeat
- Uninterruptible Application:** false
- Application Stack Label expression:** /usr/bin
- RM container Stack Label expression:** /usr/bin

Below the details, there is a section for 'Resource Usage' with the following data:

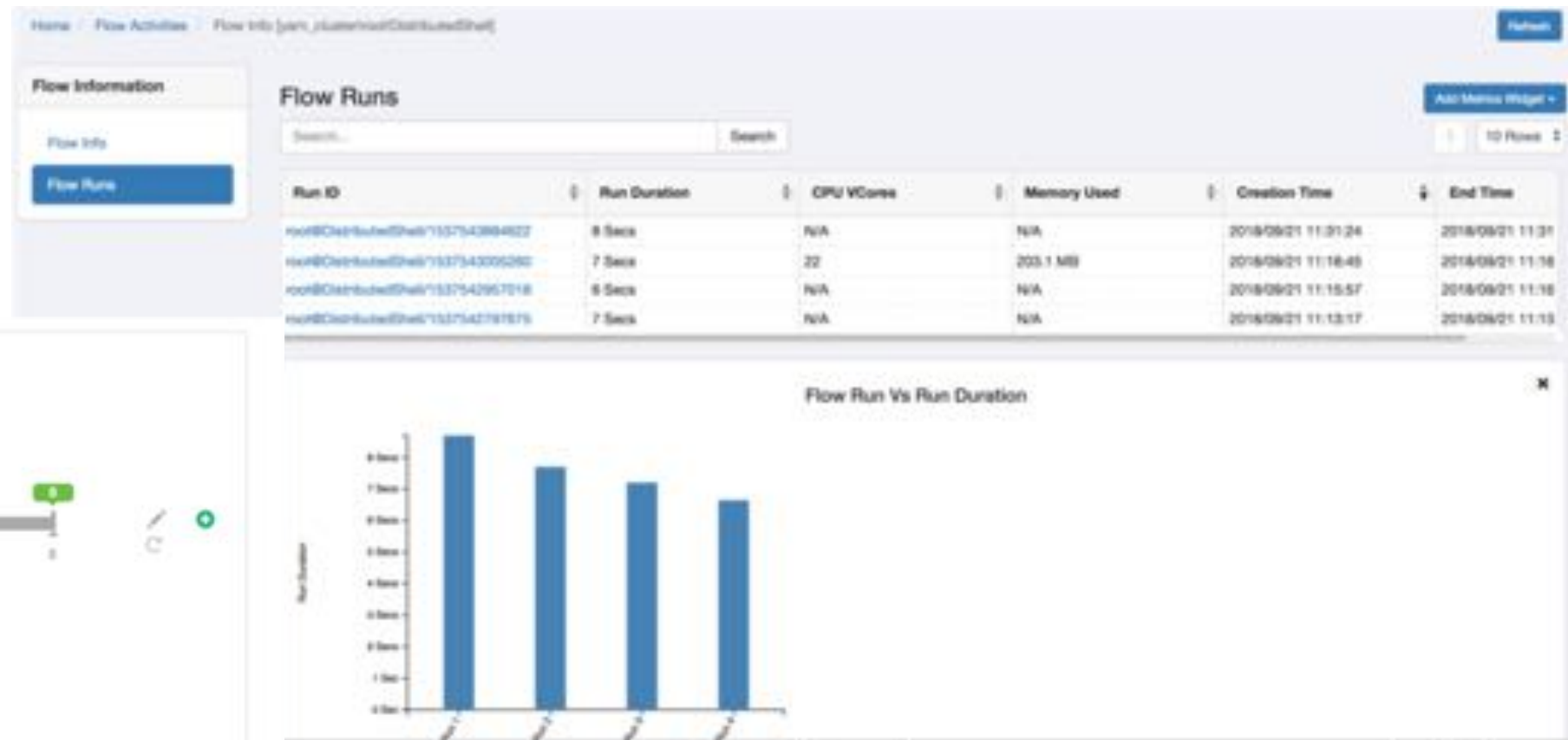
Resource	Used	Available
Total Resources Requested	1 m1	1 m1
Total Number of Non-AM Containers Provisioned	1	1
Total Number of AM Containers Provisioned	1	1
Resources Provisioned from Current Storage	1 m1	1 m1
Number of Non-AM Containers Provisioned from Current Storage	1	1
Aggregate Resource Allocation	1 m1	1 m1

<https://github.com/tspannhw/nifi-mxnet-yarn>

Uses: Python 2.7

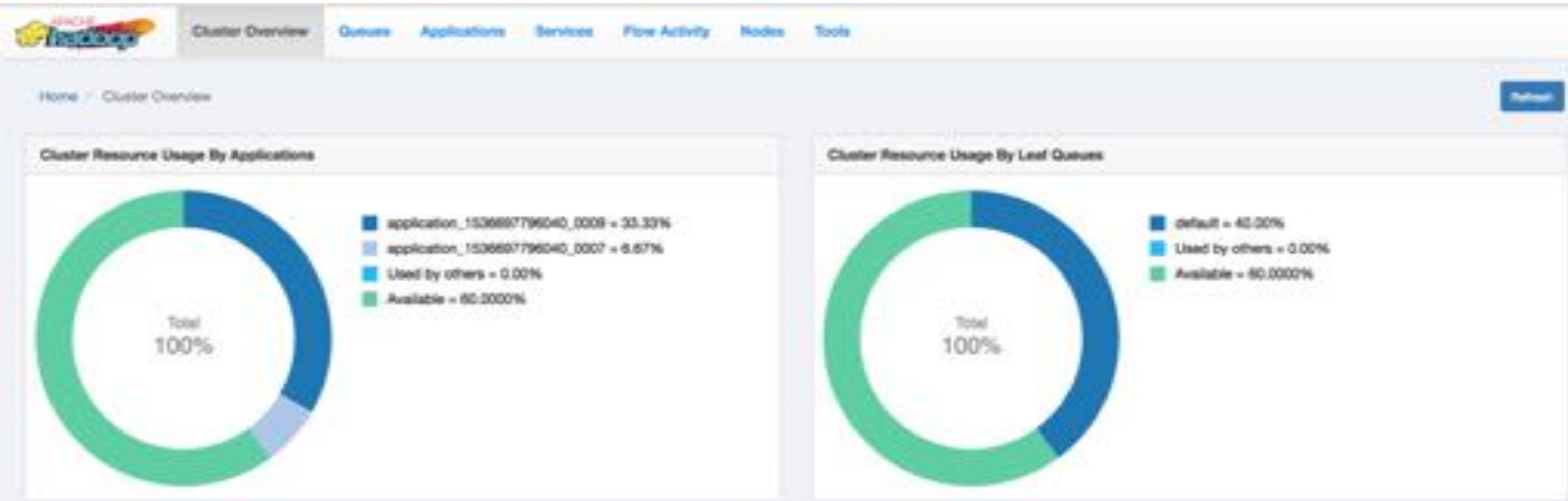
Apache MXNet on Apache YARN 3.1 Native No Spark

```
yarn jar /usr/hdp/current/hadoop-yarn-client/hadoop-yarn-applications-distributedshell.jar -jar /usr/hdp/current/hadoop-yarn-client/hadoop-yarn-applications-distributedshell.jar -shell_command python3.6 -shell_args "/opt/demo/analyzex.py /opt/images/cat.jpg" -container_resources memory-mb=512,vcores=1
```



The screenshot shows the GPU configuration interface. Under 'GPU Scheduling and Isolation', there is a 'Disabled' button. Below that, there is a text input field for the 'Absolute path of nvidia-smi on NodeManagers'. To the right, there is a 'Container' section with a 'Maximum Container Size (GPU)' slider, which is currently set to 1.

Apache MXNet on Apache YARN 3.1 Native No Spark



<https://community.hortonworks.com/content/kbentry/222242/running-apache-mxnet-deep-learning-on-yarn-31-hdp.html>

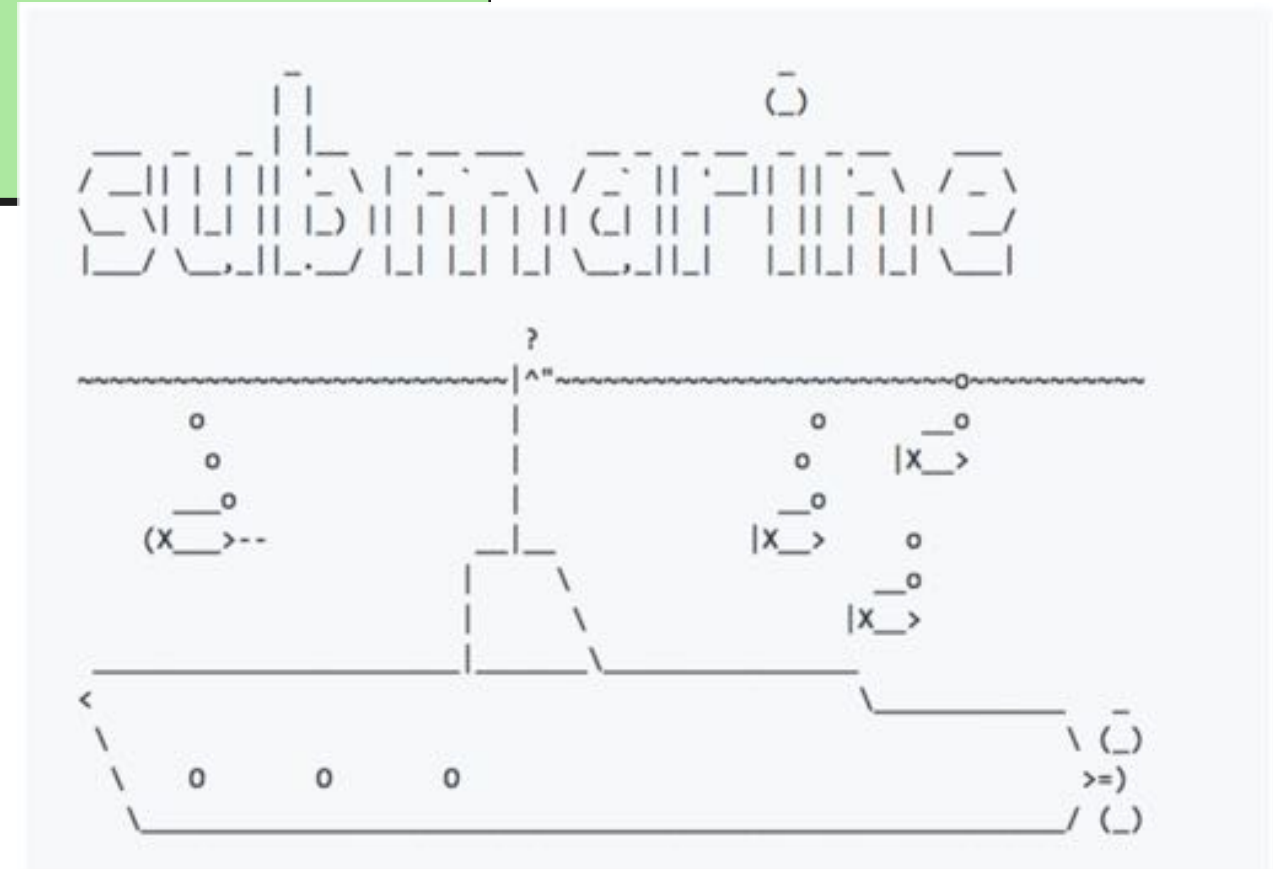
<https://github.com/tspannhw/ApacheDeepLearning101/blob/master/analyzehdfs.py>

Apache MXNet on YARN 3.2 in Docker Using “Submarine”

Hadoop {Submarine} Project: Running deep learning workloads on YARN

```
yarn jar hadoop-yarn-applications-submarine-<version>.jar job run \  
  --name xyz-job-001 --docker_image <your docker image> \  
  --input_path hdfs://default/dataset/cifar-10-data \  
  --checkpoint_path hdfs://default/tmp/cifar-10-jobdir \  
  --num_workers 1 \  
  --worker_resources memory=8G,vcores=2,gpu=2 \  
  --worker_launch_cmd "cmd for MXNet/PyTorch"
```

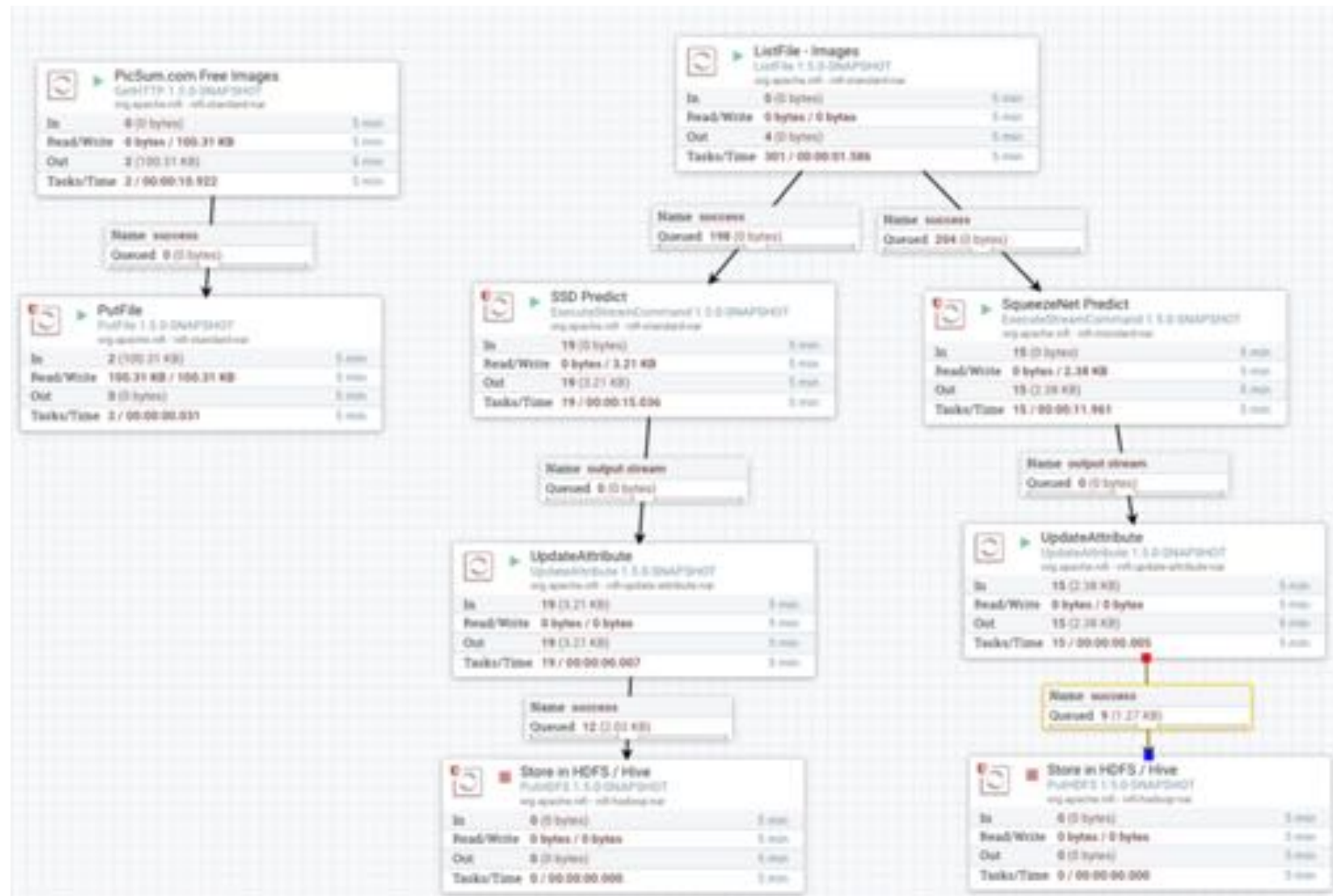
Wangda Tan (wangda@apache.org)



<https://github.com/apache/hadoop/tree/trunk/hadoop-yarn-project/hadoop-yarn/hadoop-yarn-applications/hadoop-yarn-submarine>

<https://issues.apache.org/jira/browse/YARN-8135>

Apache MXNet Model Server with Apache NiFi



<https://community.hortonworks.com/articles/155435/using-the-new-mxnet-model-server.html>

Apache MXNet Model Server with Apache NiFi

```
mxnet-model-server --models squeezeenet=https://s3.amazonaws.com/model-server/models/squeezeenet\_v1.1/squeezeenet\_v1.1.model --service mms/model_service/mxnet\_vision\_service.py --port 9999
```

```
mxnet-model-server --models SSD=resnet50_ssd_model.model --service ssd\_service.py --port 9998
```

```
View as: original
1 {
2   "prediction": [
3     [
4       {
5         "class": "n02825657 bell cote, bell cot",
6         "probability": 0.49351149797439575
7       },
8       {
9         "class": "n04366367 suspension bridge",
10        "probability": 0.17974209785461426
11      },
12      {
13        "class": "n03028079 church, church building",
14        "probability": 0.11694391071796417
15      },
16      {
17        "class": "n03032252 cinema, movie theater, movie theatre, movie house, picture palace",
18        "probability": 0.07838434725999832
19      },
20      {
21        "class": "n03781244 monastery",
22        "probability": 0.04639515280723572
23      }
24    ]
25  }
26 }
27
```

<https://community.hortonworks.com/articles/177232/apache-deep-learning-101-processing-apache-mxnet-m.html>

Apache OpenNLP with Apache NiFi

Apache OpenNLP for Entity Resolution Processor

<https://github.com/tspannhw/nifi-nlp-processor>

Requires installation of NAR and Apache OpenNLP Models (<http://opennlp.sourceforge.net/models-1.5/>).

This is a non-supported processor that I wrote and put into the community. You can write one too!

<https://community.hortonworks.com/articles/80418/open-nlp-example-apache-nifi-processor.html>

<https://opennlp.apache.org/news/release-190.html>



FlowFile

DETAILS ATTRIBUTES

Attribute Values

filename
2788601463132800.json

names
{ "names": [{"name": "Tim Spann"}, {"name": "Peter Smith"}] }

followers_count
47

location
Columbus, Ohio

locations
{ "locations": [{"location": "Sydney"}] }



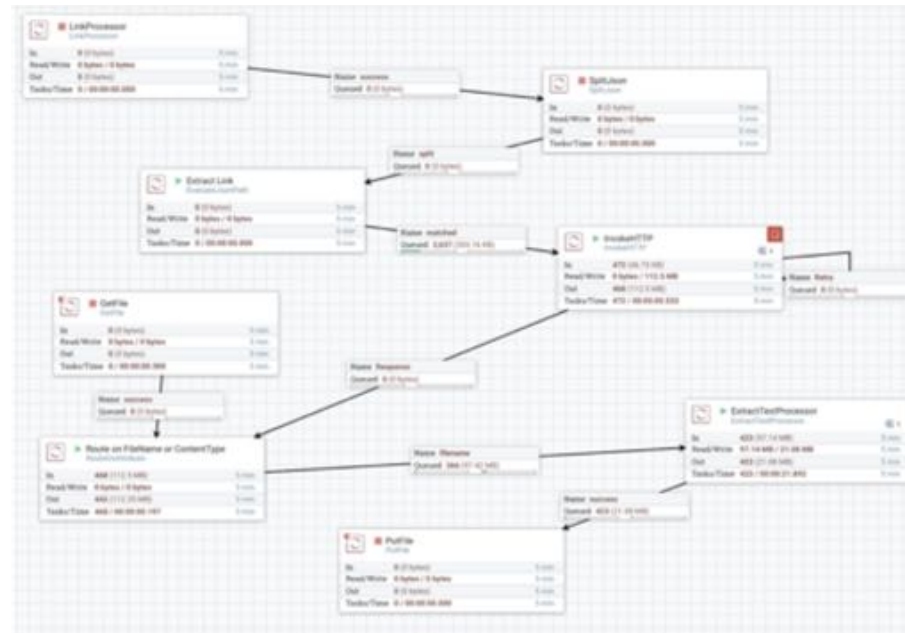
Apache Tika with Apache NiFi



Displaying 1 of 195

Type	Tags
ExtractTextProcessor	extracttextprocessor

Selected Processor:
ExtractTextProcessor
Run Tika Text Extraction from PDF, Word, Excel



```
View: original
40 streaming data. The combination of messaging and processing technologies enables stream
41 processing at linear scale.
42 For example, Apache Storm ships with support for Kafka as a data source using Storm's core
43 API or the higher-level, micro-batching Trident API. Storm's Kafka integration also includes
44 support for writing data to Kafka, which enables complex data flows between components in a
45 Hadoop-based architecture. For more information about Apache Storm, see the Storm User Guide.
46
47
48 Legal notices
49
50 Contents
51 Search
52
53 1. Building a High-Throughput Messaging System with Apache Kafka
54 2. What's New 1. Apache Kafka
55 2. Content Updates
56
57
58 3. Apache Kafka Concepts
59 4. Installing Kafka 1. Prerequisites
60 2. Installing Kafka Using Ambari
61
62
63 5. Configuring Kafka for a Production Environment 1. Customizing Kafka Configuration Settings 1.1. Connection Settings
64 1.2. Topic Settings
65 1.3. Log Settings
66 1.4. Compaction Settings
67 1.5. Advanced kafka-env Settings
68 1.6. Adding Configuration Properties
69
70
71 2. Configuring ZooKeeper for Multiple Applications
72 3. Enabling Audit to HDFS for a Secure Cluster
73
74
75
76 6. Mirroring Data Between Clusters: Using the MirrorMaker Tool 1. Running MirrorMaker
77 2. Checking Mirroring Progress
78 3. Avoiding Data Loss
79 4. Running MirrorMaker on Kerberos-Enabled Clusters
80
81
82 7. Developing Kafka Producers and Consumers
83
84 Search
```

<https://github.com/tspannhw/nifi-extracttext-processor>

<https://community.hortonworks.com/articles/76924/data-processing-pipeline-parsing-pdfs-and-identify.html>

<https://community.hortonworks.com/articles/81694/extracttext-nifi-custom-processor-powered-by-apach.html>

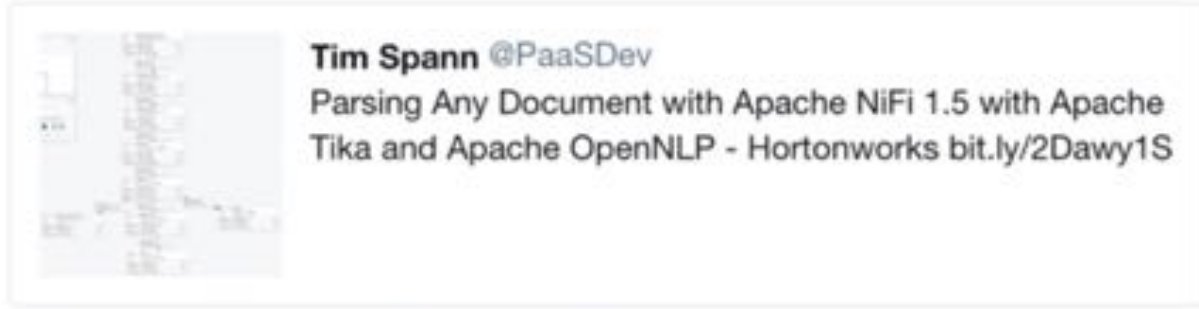
<https://community.hortonworks.com/articles/163776/parsing-any-document-with-apache-nifi-15-with-apac.html>

<https://community.hortonworks.com/content/kbentry/177370/extracting-html-from-pdf-excel-and-word-documents.html>

Another Reason Apache and Apache Tika are Awesome!



Tim Allison @_tallison · Mar 30
@PaaSDev this is fantastic! Thank you! W00t @apachenifi @ApacheTika and @ApacheOpennlp !



2 replies 2 retweets 3 likes

Tim Allison @_tallison Following

Happy to do code review of the @ApacheTika bits if you'd like. Tika 1.18 is in the pre release process now.

3:35 PM - 30 Mar 2018

2 Likes

2 replies 2 likes

<https://community.hortonworks.com/articles/163776/parsing-any-document-with-apache-nifi-15-with-apac.html>

<https://github.com/tspannhw/nifi-extracttext-processor>

Contact

<https://twitter.com/PaaSDev>

<https://github.com/tspannhw/apache-deep-learning-101>

<https://community.hortonworks.com/users/9304/tspann.html>

<https://dzone.com/users/297029/bunkertor.html>

<https://www.meetup.com/futureofdata-princeton/>

<http://gluon-crash-course.mxnet.io/>

<https://community.hortonworks.com/articles/155435/using-the-new-mxnet-model-server.html>

<https://github.com/dmlc/dmlc-core/tree/master/tracker/yarn>

<https://unsplash.com/> <https://pixabay.com/>

