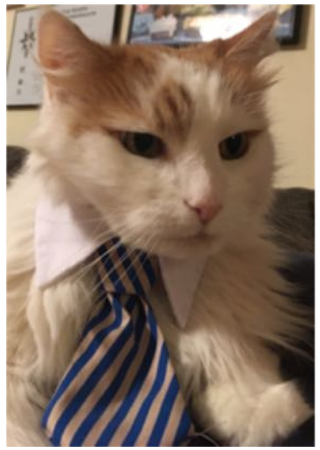
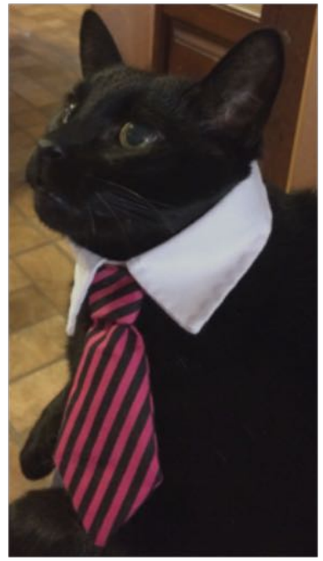
Apache Deep Learning 101 v0.31
(For Data Engineers)




```
{
  "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/ApacheDeepLearning101/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

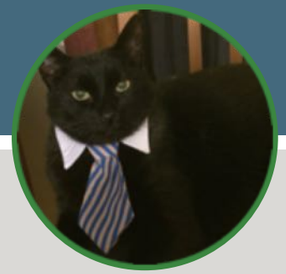
@PaaSDev

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



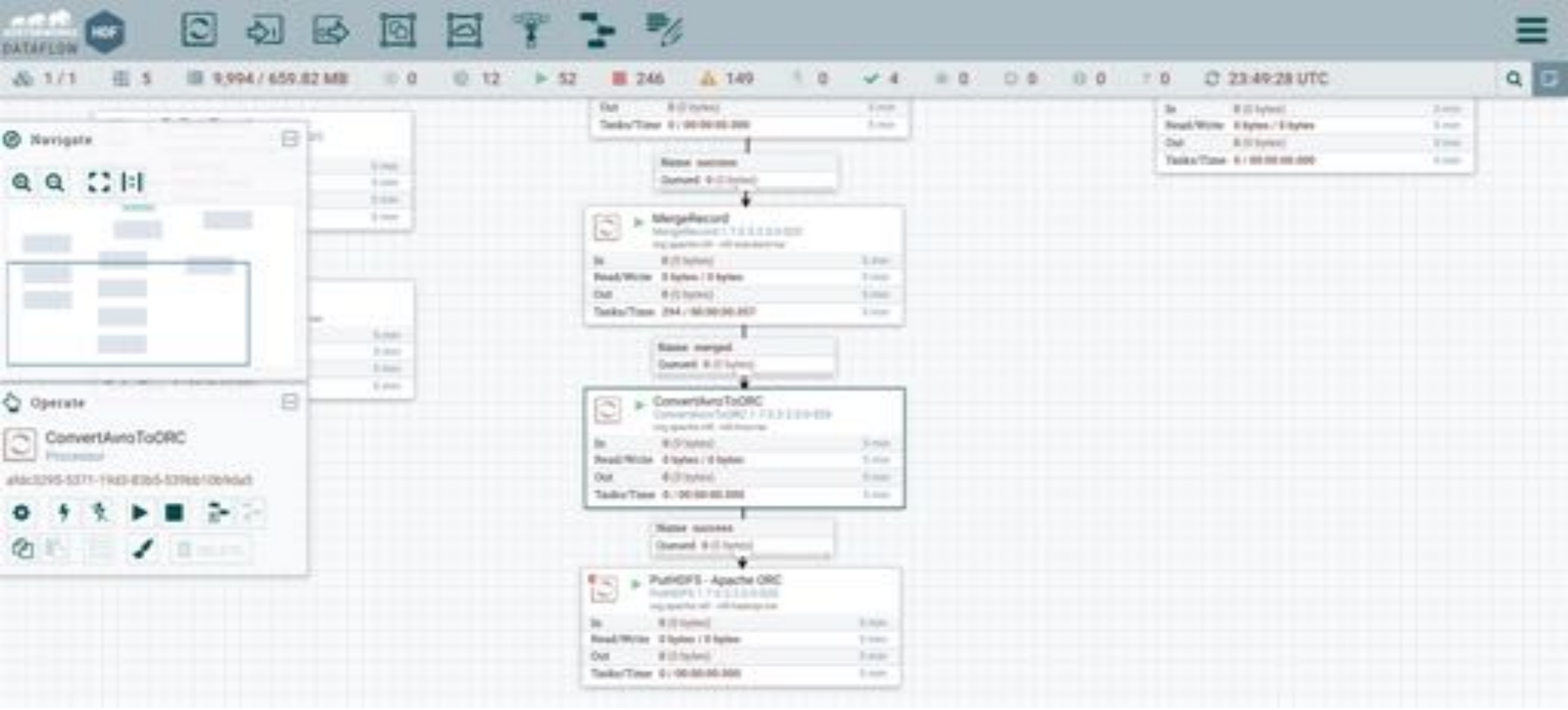
CAT

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

AI

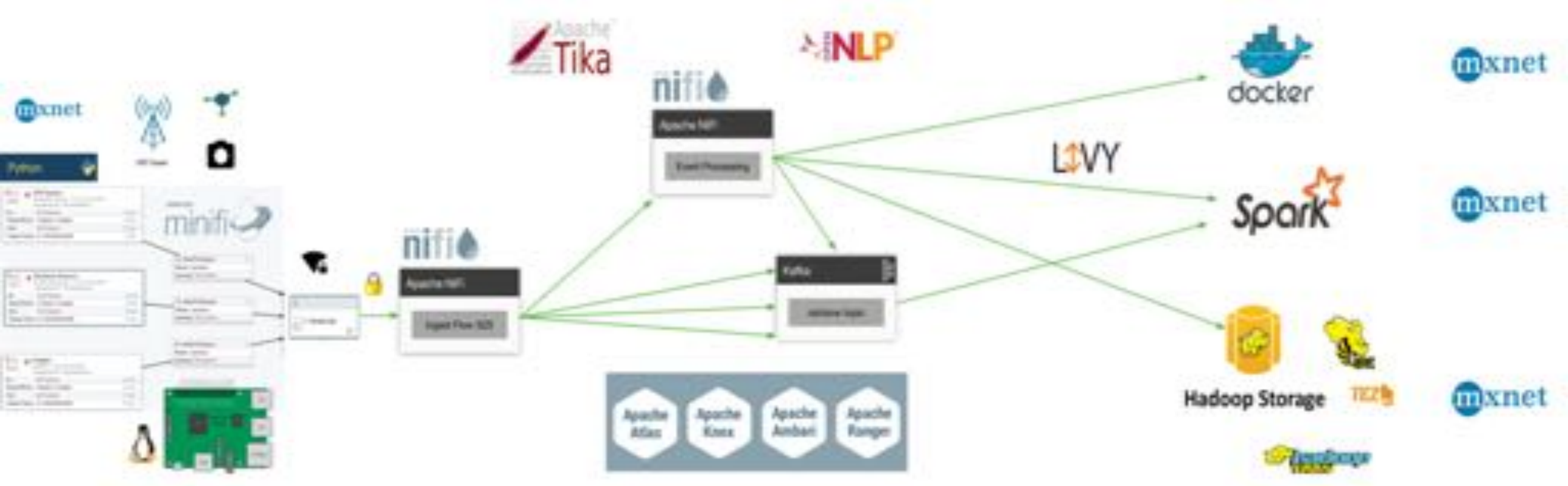
- Will Drive your Car
- Will Fix Your Code
- Will Beat You At Q-Bert
- Will Not Be Discussed Today
- Will Not Finish This Talk For Me, This Time

http://gluon.mxnet.io/chapter01_crashcourse/preface.html


DATAFLOW
Navigate
Operate
ConvertAvroToORC
MergeRecord
ConvertAvroToORC
PutHDFS - Apache ORC

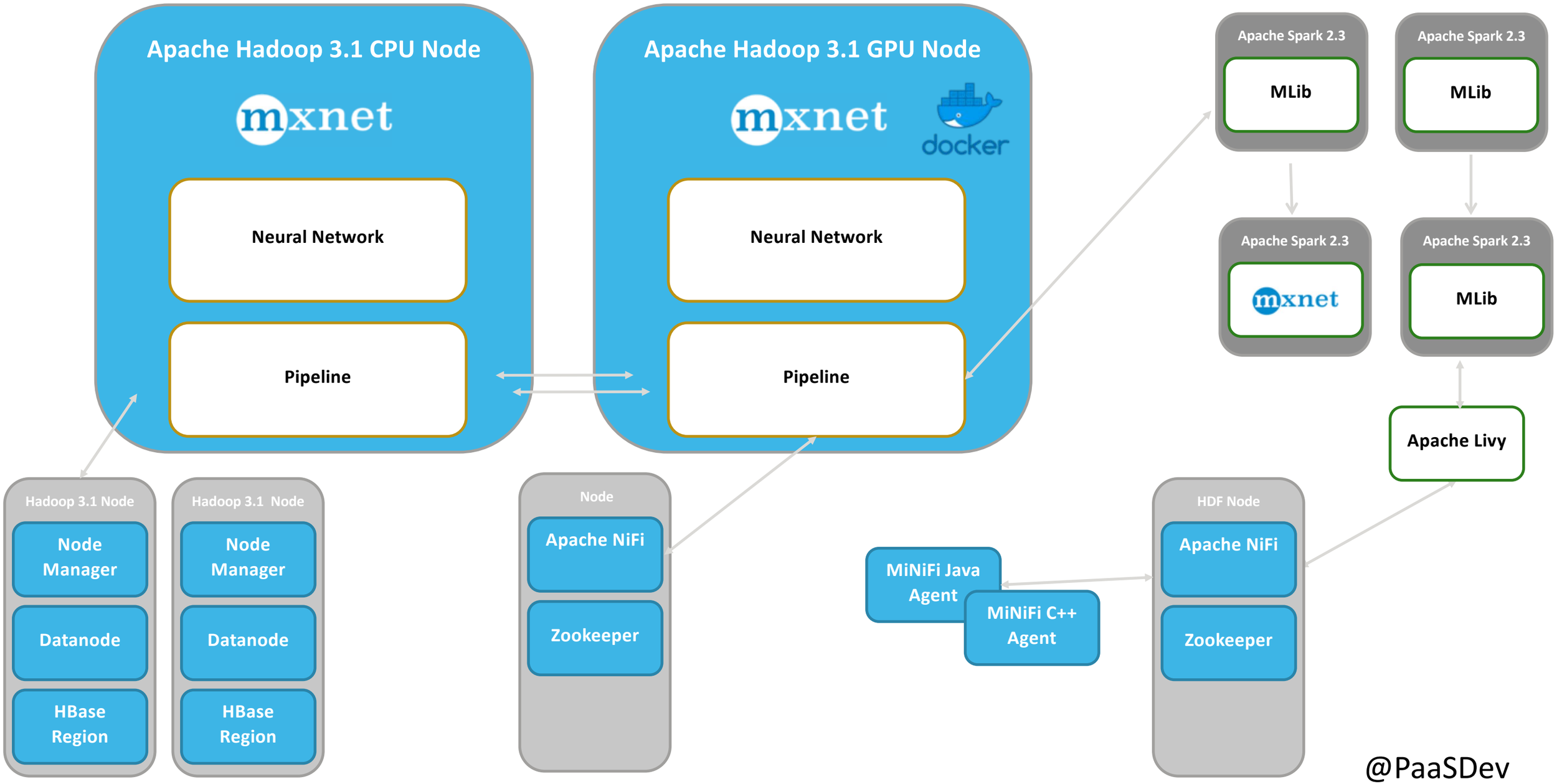
@PaaSDev

Apache Deep Learning Flow



@PaaSDev

Apache Deep Learning Server Architecture



@PaaSDev

Collect: Bring Together

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

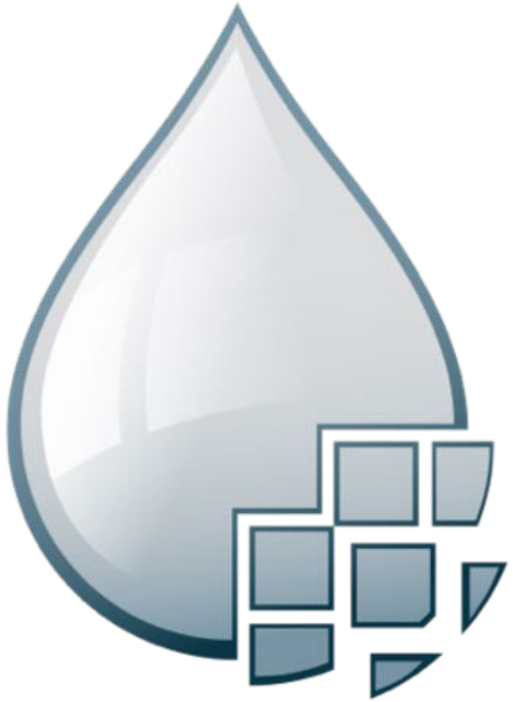
Conduct: Mediate the Data Flow

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

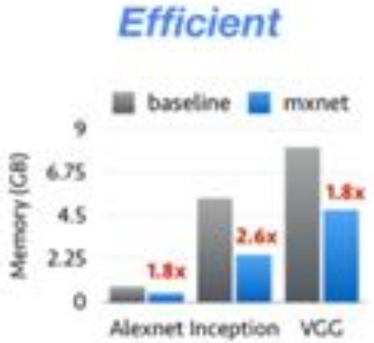
- Guaranteed delivery
- Data buffering
  - Backpressure
  - Pressure release
- Prioritized queuing
- Flow specific QoS
  - Latency vs. throughput
  - Loss tolerance
- Data provenance

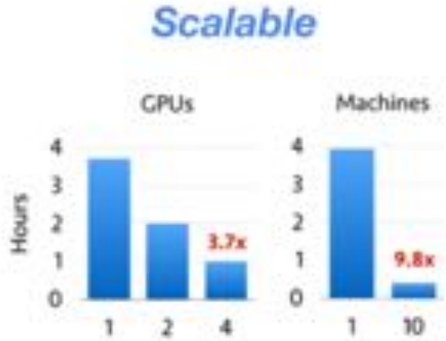
- Recovery / recording a rolling log of fine-grained history
- Designed for extension

Different from Apache NiFi

- **Design and Deploy**
- **Warm re-deploys**

mxnet





- 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

mxnet

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

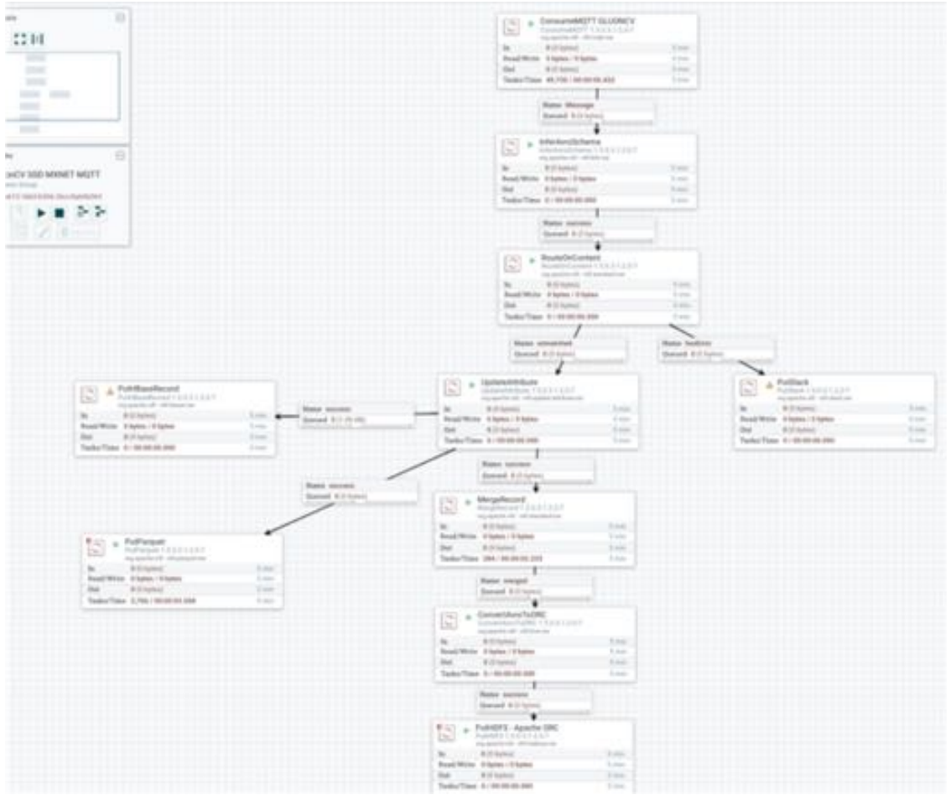
Apache MXNet via Python (OSX Local with WebCam)

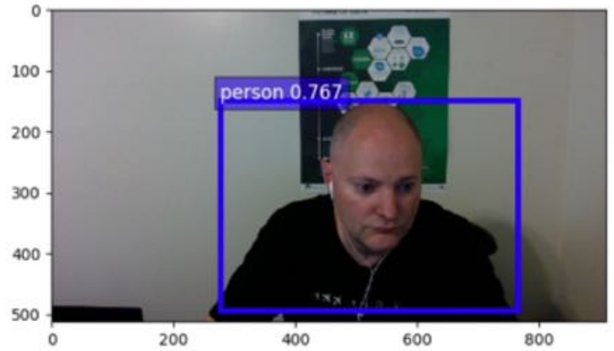
python3 -W ignore analyze.py

{"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





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

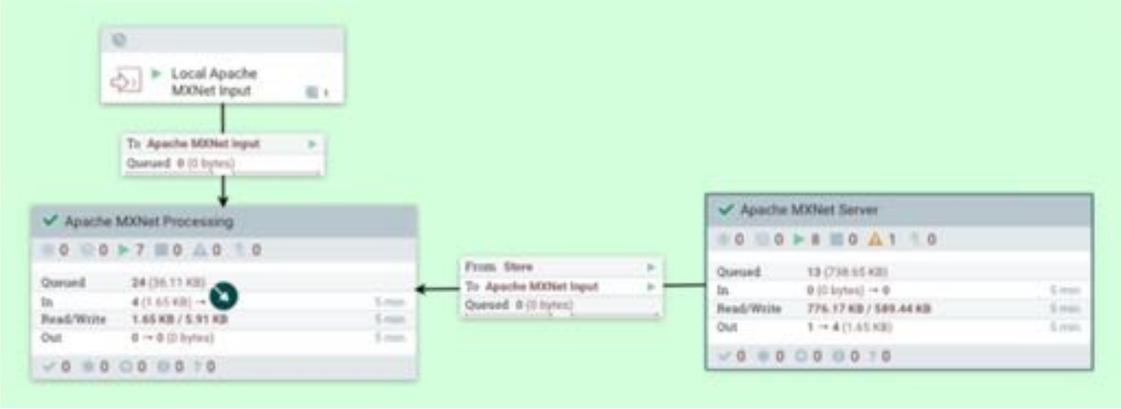
Apache MXNet Installation on OSX

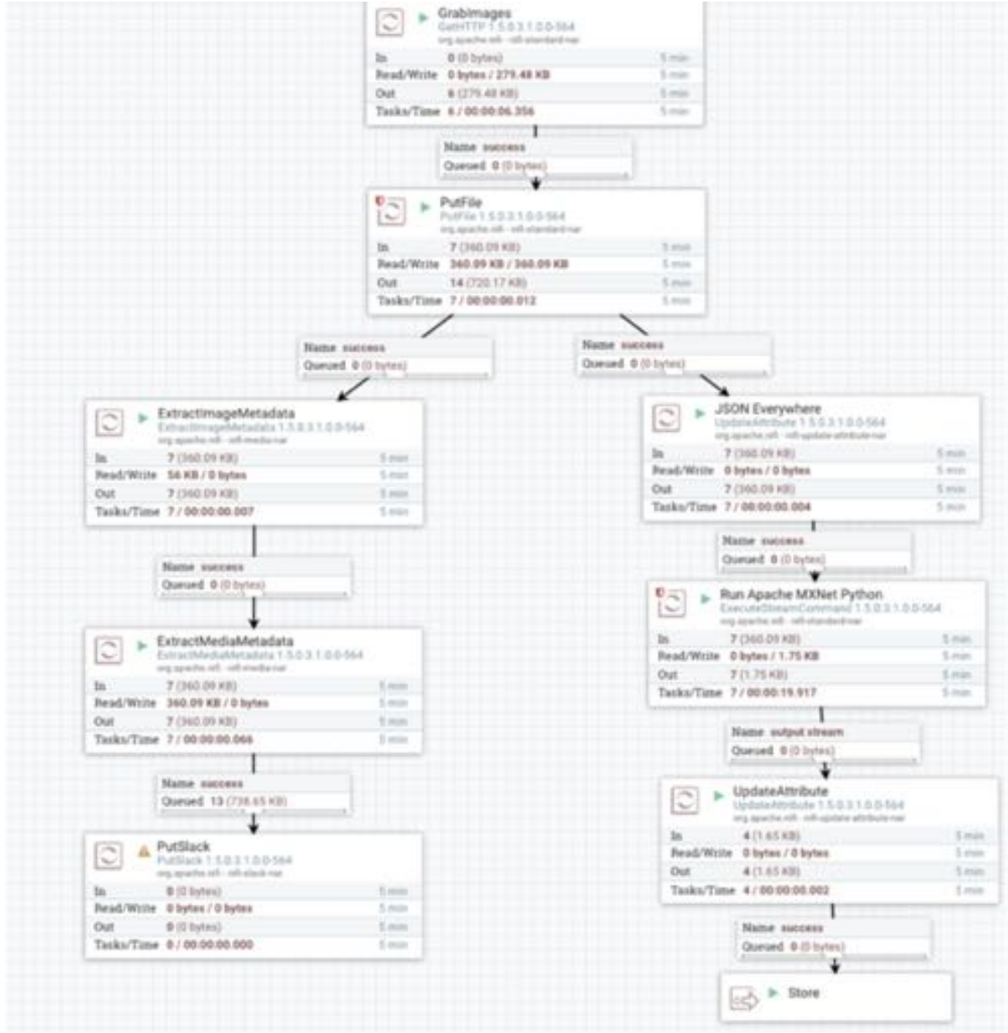
```
git clone https://github.com/apache/incubator-mxnet.git
cd incubator-mxnet
mkdir images
curl --header 'Host: data.mxnet.io' --header 'User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.11; rv:45.0) Gecko/20100101 Firefox/45.0' --header 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8' --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





@PaaSDev

Apache MXNet Installation on an Centos 7

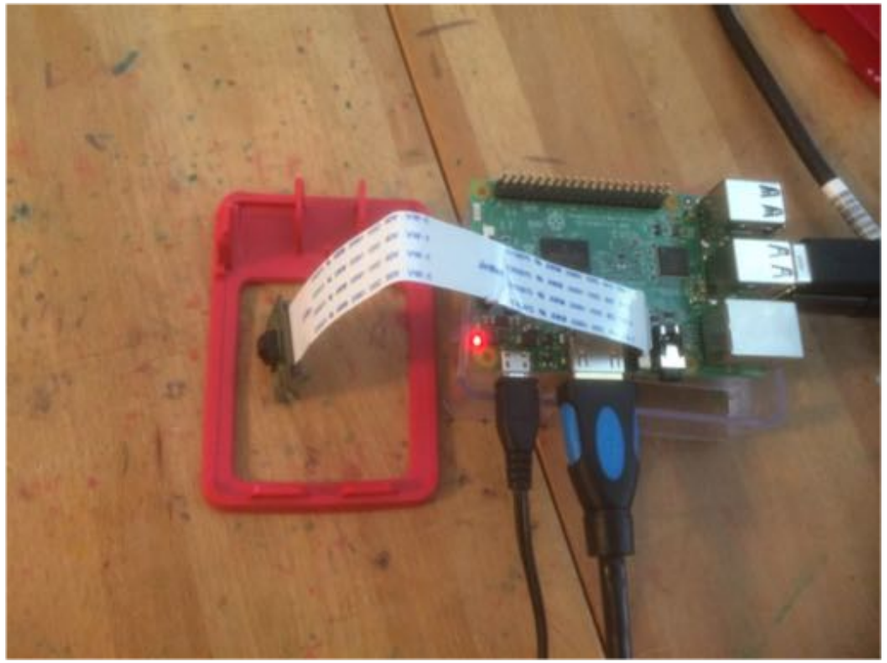
git clone https://github.com/apache/incubator-mxnet.git

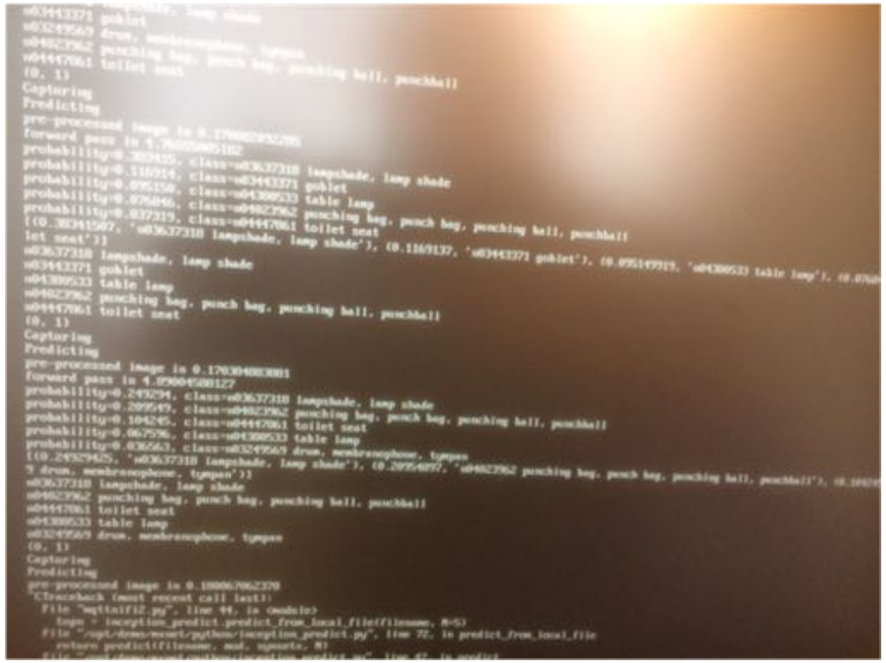
```
sudo yum groupinstall 'Development Tools' -y
sudo yum install cmake git pkgconfig -y
sudo yum install libpng-devel libjpeg-turbo-devel jasper-devel openexr-devel
libtiff-devel libwebp-devel -y
sudo yum install libdc1394-devel libv4l-devel gstreamer-plugins-base-devel -y
sudo yum install gtk2-devel -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)





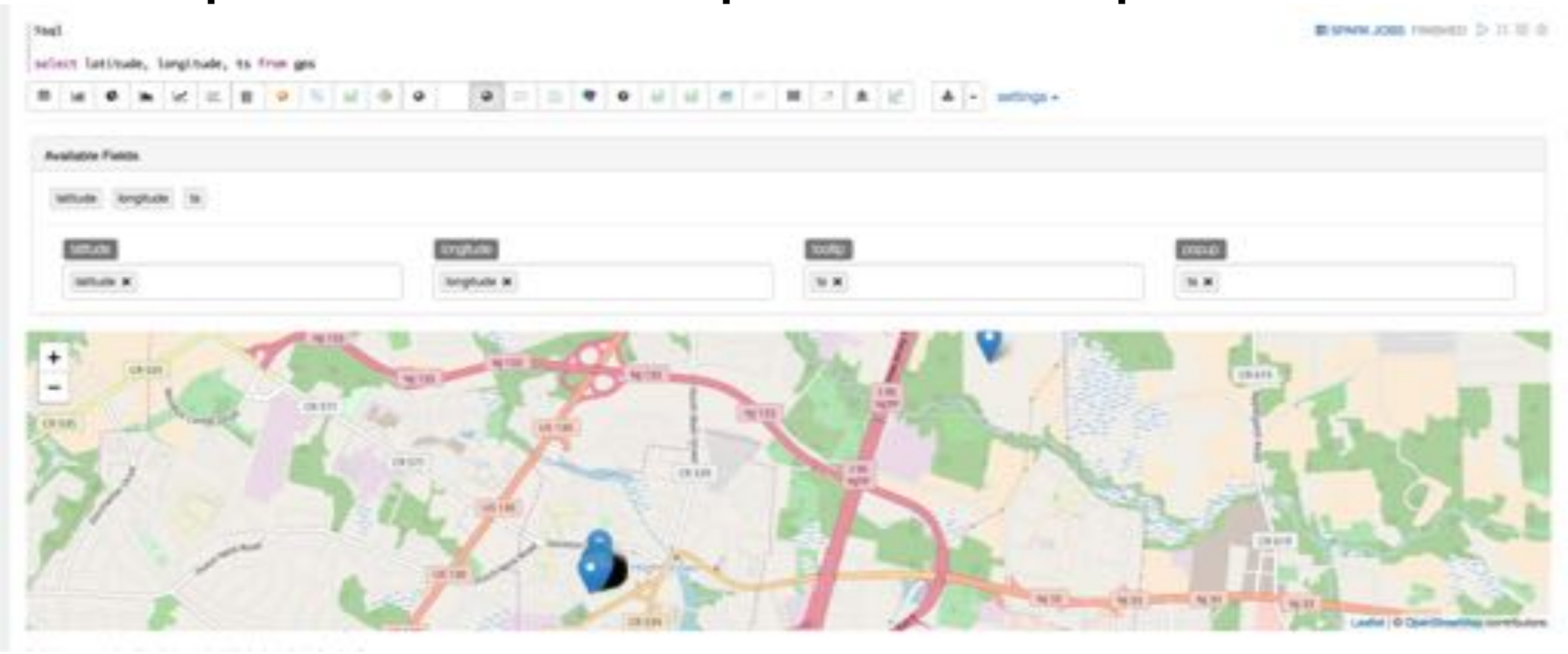
https://github.com/tspannhw/OpenSourceComputerVision

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

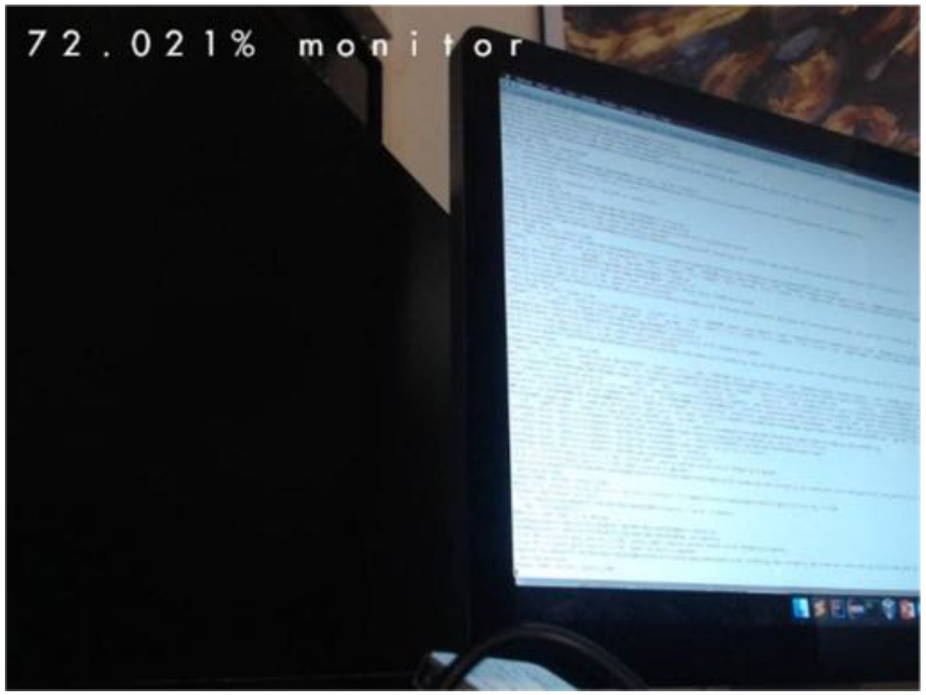
Multiple IoT Devices with Apache NiFi and Apache MXNet

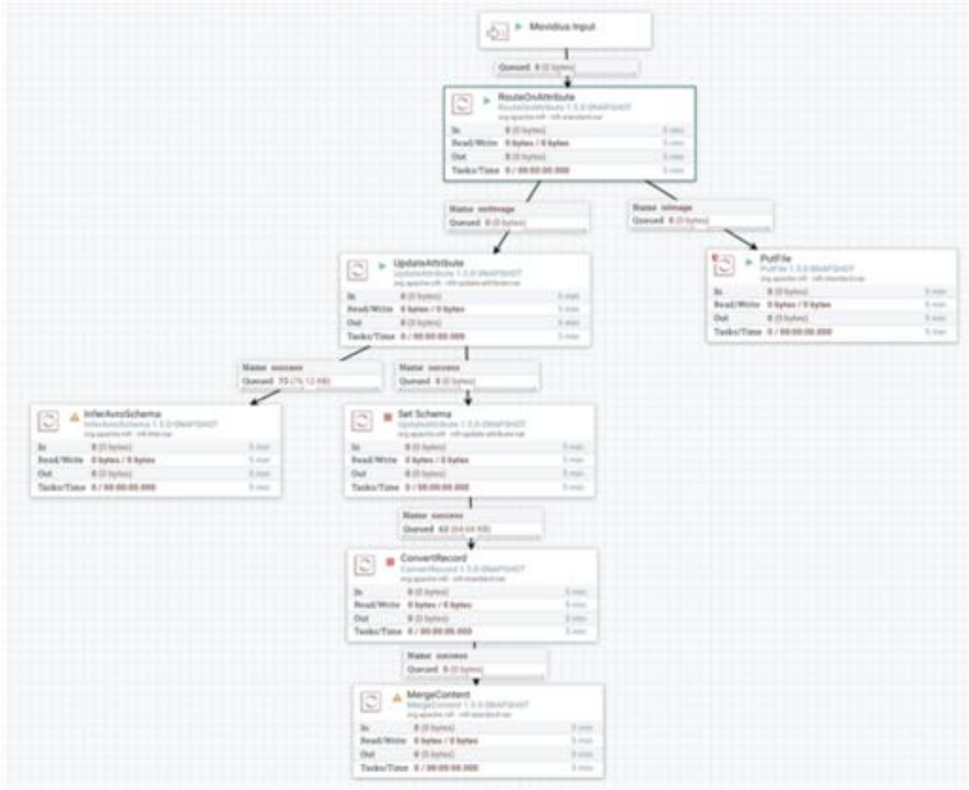


https://community.hortonworks.com/articles/203638/ingesting-multiple-iot-devices-with-apache-nifi-17.html

@PaaSDev

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 Running in Apache Zeppelin

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

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

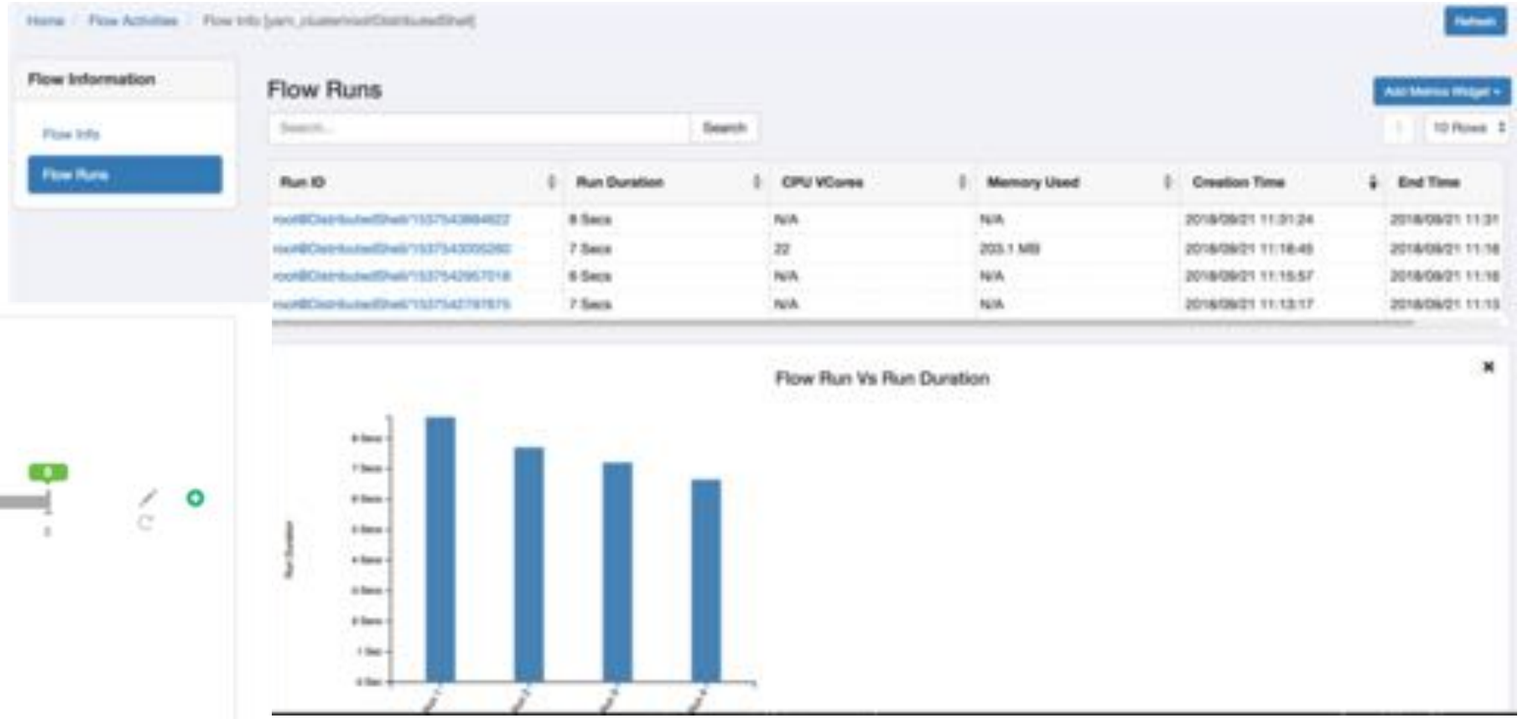
```
dmlc-submit --cluster yarn --num-workers 1 --server-cores 2
--server-memory 1G --log-level DEBUG --log-file mxnet.log analyzeyarn.py
```

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

Uses: Python 2.7

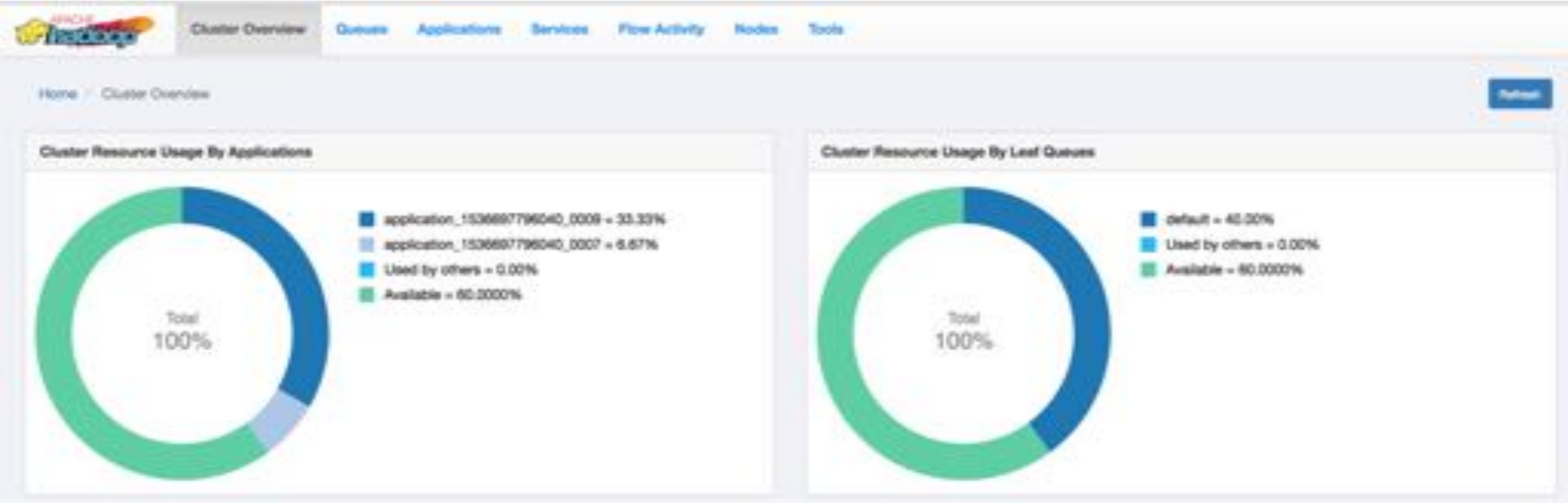
Apache MXNet on Apache YARN 3.1 Native No Spark

```
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
```



Uses: Python Any

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

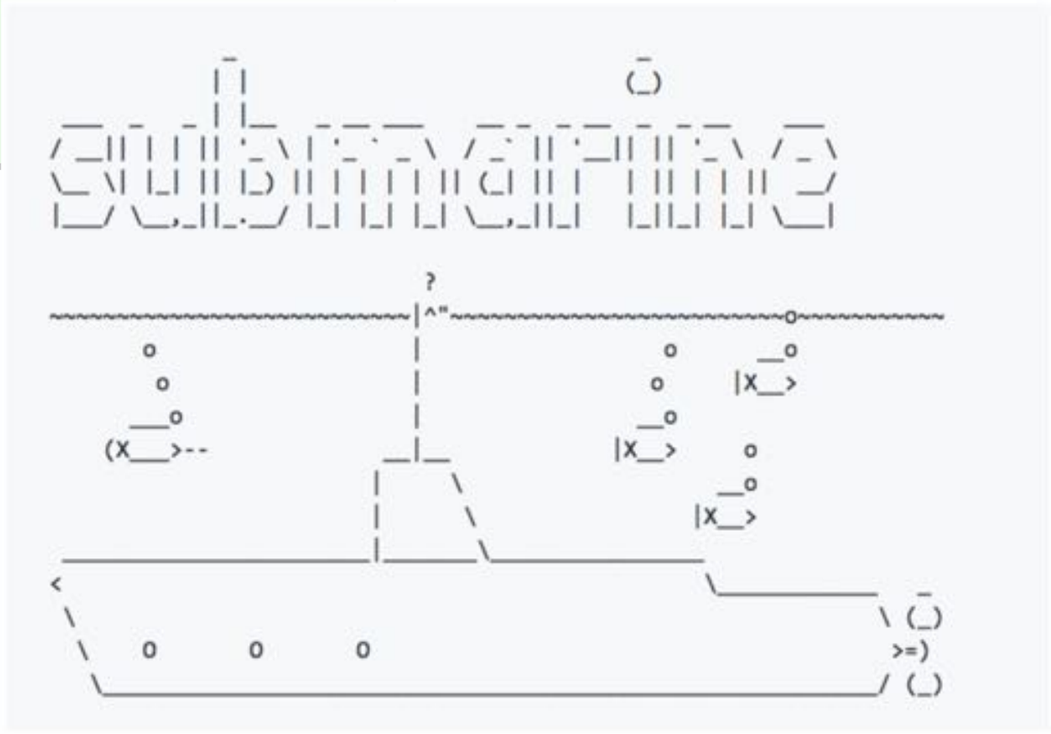
https://github.com/tspannhw/ApacheDeepLearning101/blob/master/analyzehdfs.py

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"
```

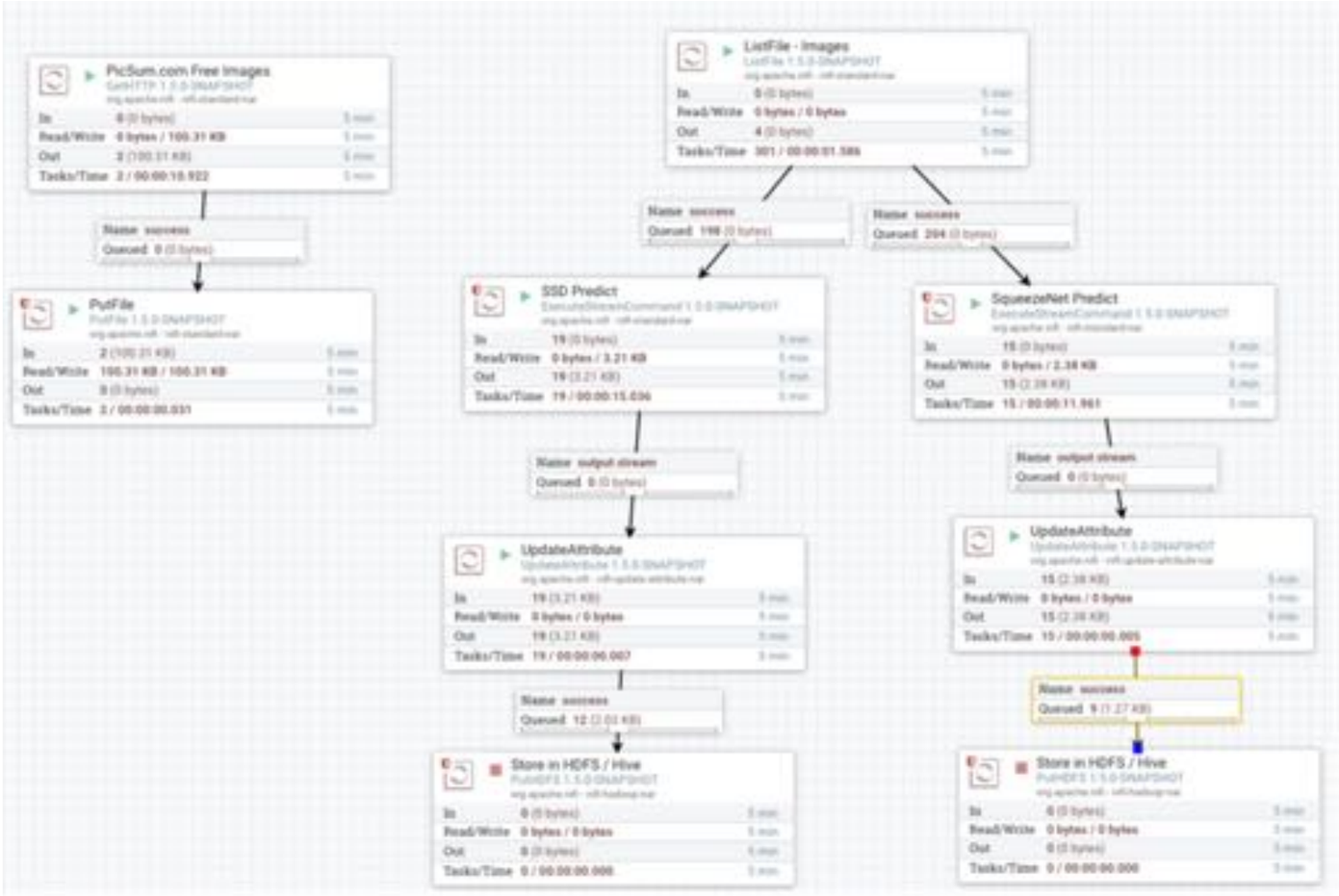
Wangda Tan (wangda@apache.org)



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



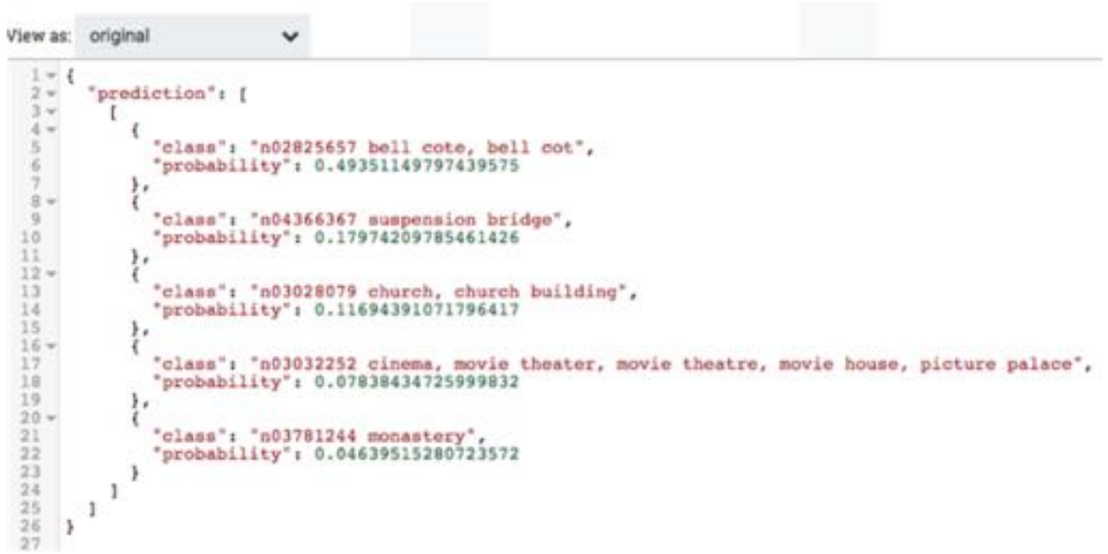
https://community.hortonworks.com/articles/155435/using-the-new-mxnet-model-server.html

sudo pip3 install mxnet-model-server --upgrade

Apache MXNet Model Server with Apache NiFi

mxnet-model-server --models squeezenet=https://s3.amazonaws.com/model-server/models/squeezenet_v1.1/squeezenet_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



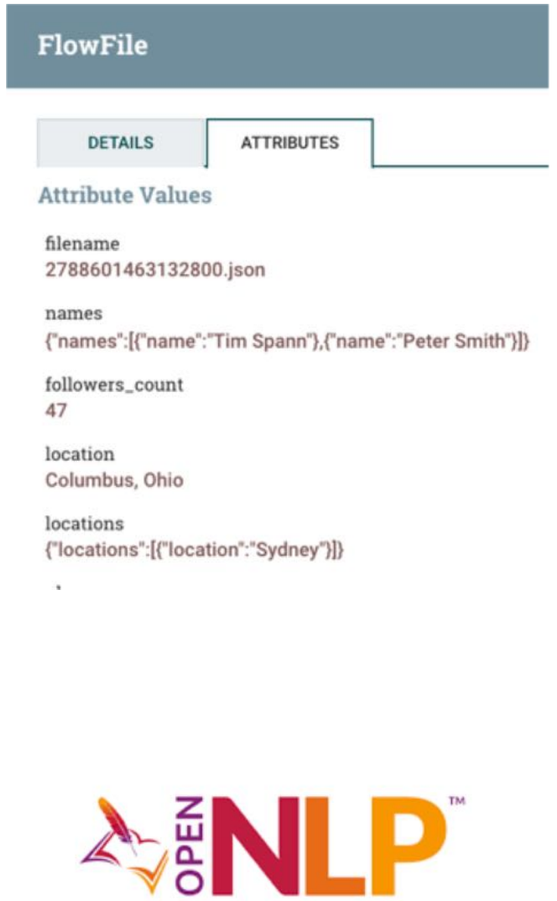
https://community.hortonworks.com/articles/177232/apache-deep-learning-101-processing-apache-mxnet-m.html

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!

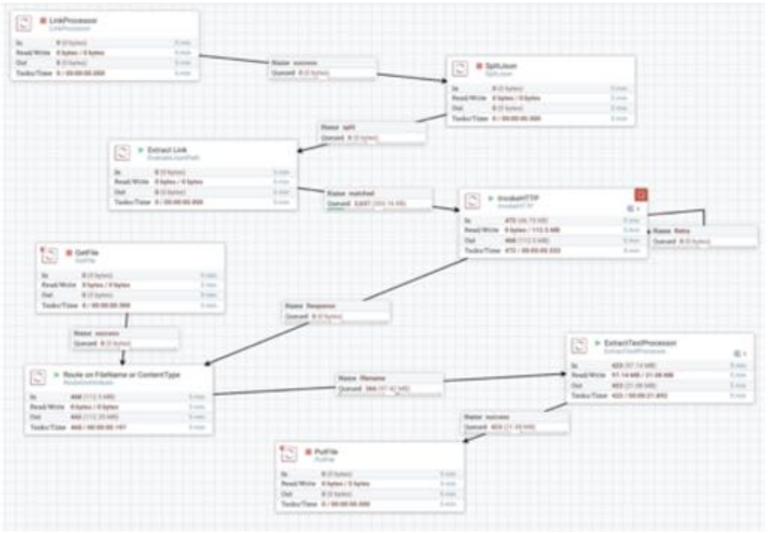


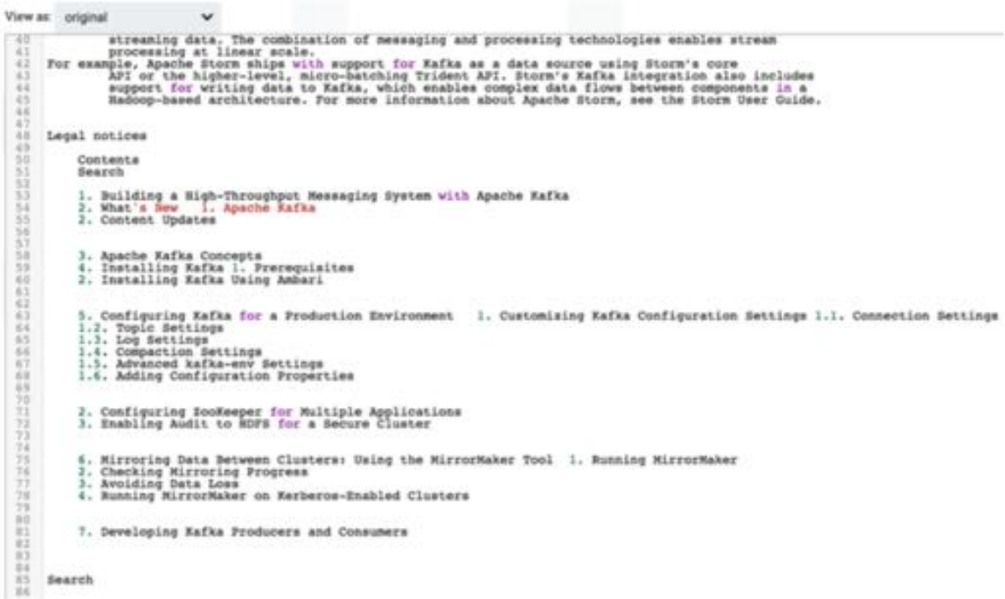
https://community.hortonworks.com/articles/80418/open-nlp-example-apache-nifi-processor.html

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

@PaaSDev

Apache Tika with Apache NiFi





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

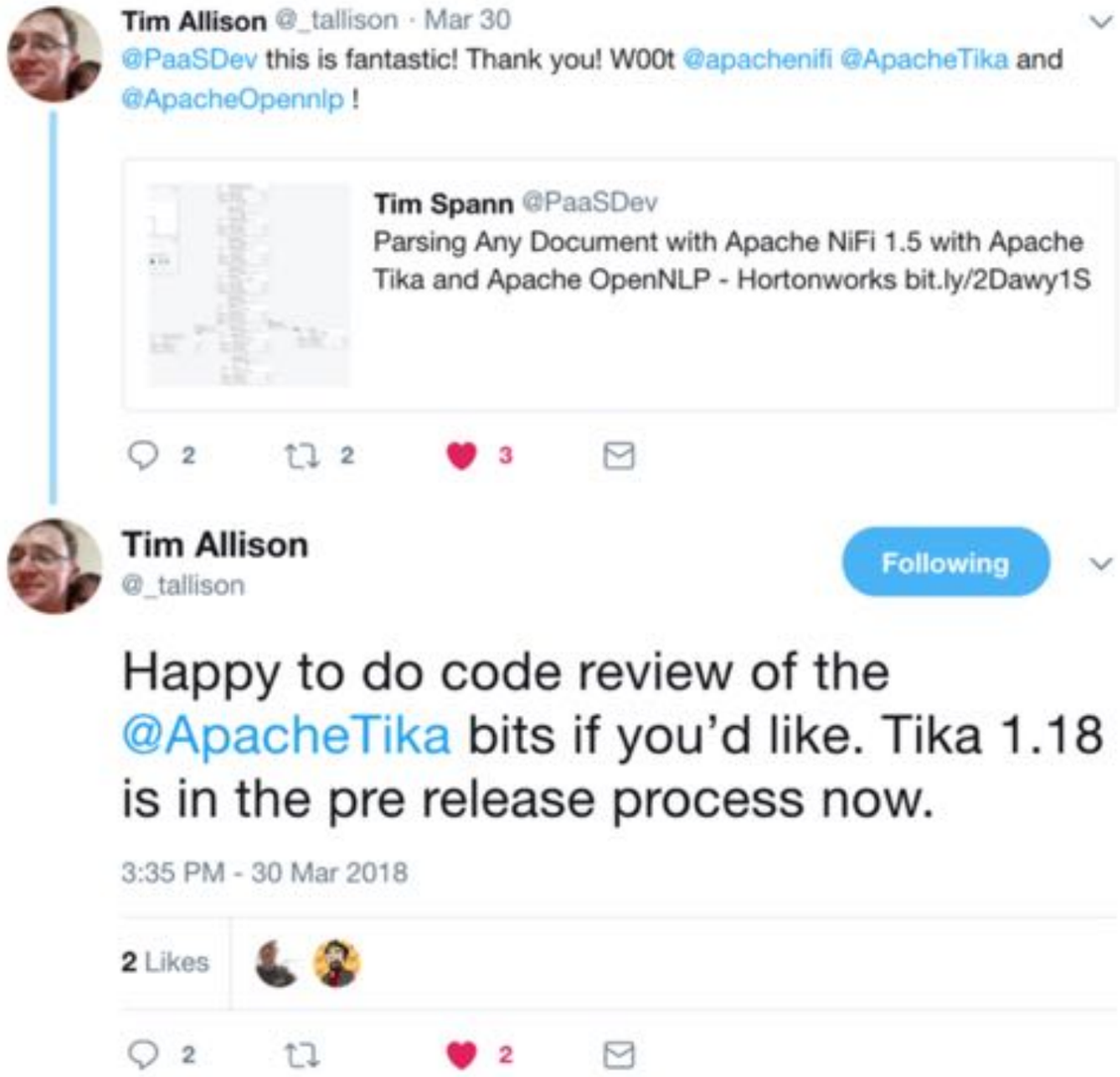
https://community.hortonworks.com/articles/76924/data-processing-pipeline-parsing-pdfs-and-identify.html

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!



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

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

@PaaSDev

Contact

https://twitter.com/PaaSDev

https://github.com/tspannhw/ApacheDeepLearning101

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/