

Hail Hydrate! From Stream to Lake

Tim Spann

Principal DataFlow Field Engineer
@ Cloudera

Hail Hydrate!
From Stream to Lake

Conf42: Machine Learning 2021
Thursday July 29 | 5PM GMT



Timothy Spann
Developer Advocate

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

Tim

SPANNN

<https://github.com/tspannhw>

<https://www.datainmotion.dev/>

Speaker Bio

Developer Advocate

DZone Zone Leader and Big Data MVB;
@PaasDev

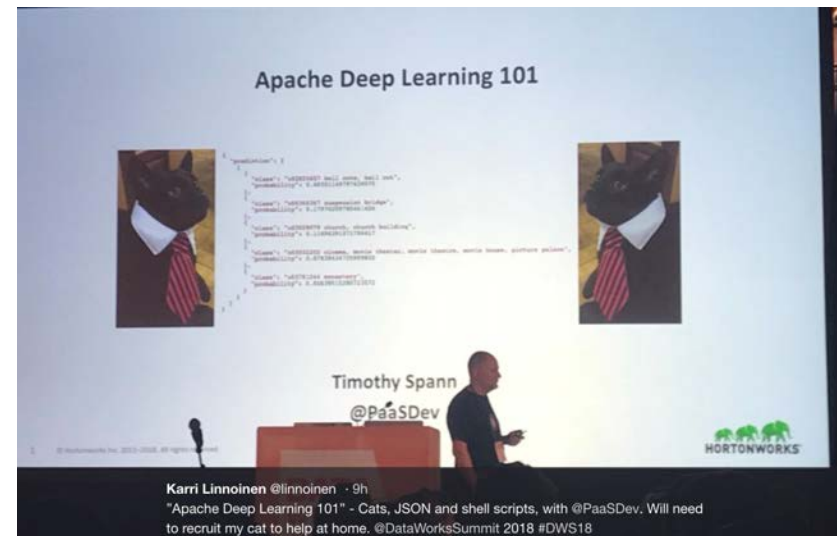
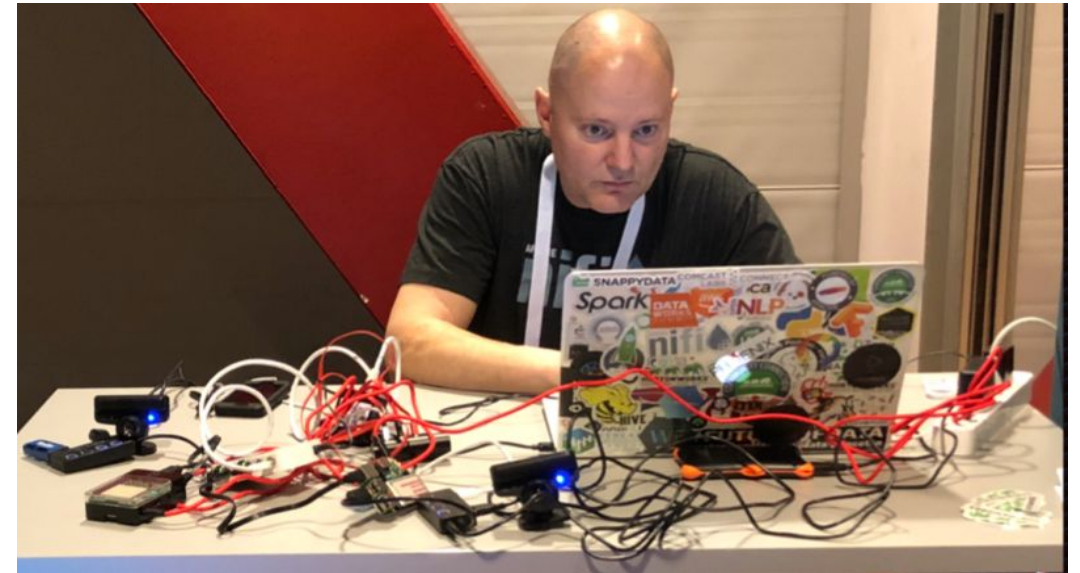
<https://github.com/tspannhw> <https://www.datainmotion.dev/>

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

<https://dev.to/tspannhw>

<https://sessionize.com/tspann/>

<https://www.slideshare.net/bunkertor>



AGENDA



Use Case - Populate the Data Lake

Key Challenges

- Their Impact
- A Solution
- Outcome

Why Apache NiFi and Apache Pulsar?

Successful Architecture

Demo

Next Steps



USE CASE

IoT Ingestion: High-volume streaming sources, multiple message formats, diverse protocols and multi-vendor devices creates data ingestion challenges.





KEY CHALLENGES



Data Ingestion: High-volume streaming sources, multiple message formats, diverse protocols and multi-vendor devices creates data ingestion challenges.



Real-time Insights: Analyzing continuous and rapid inflow (velocity) of streaming data at high volumes creates major challenges for gaining real-time insights.



Visibility: Lack visibility of end-to-end streaming data flows, inability to troubleshoot bottlenecks, consumption patterns etc.



IMPACT



Code Sprawl: Custom scripts over various qualities proliferate across environments to cope with the complexity.



Costs: Increasing costs of development and maintenance. Too many tools, not enough experts, waiting for contractors or time delays as developers learn yet another tool, package or language.



Delays: Decreasing user satisfaction and delay in project delivery. Missed revenue and opportunities.



SOLUTION



Data Ingestion: Apache NiFi is the one tool handle high-volume streaming sources, multiple message formats, diverse protocols and multi-vendor devices.



Variety of Data: Apache NiFi offers hundreds of OOTB connectors and a GUI that accelerates flow developments. With Record Processors that convert types in a single fast step.



Visibility: Apache NiFi provenance provides insights, metrics and control over the entire end-to-end stream across clouds.



OUTCOME



New Applications: Enablement of new innovative use cases in compressed timeframe. No more waiting for data to arrive, Data Analysts and Data Scientists focus on innovation.



Savings: Cost reduction thanks to technologies offload, reduced consultant costs and simplification of ingest processes.



Agility: Reduction of new data source onboarding time from weeks to days. More data in your data warehouse now.



FLiP Stack for Cloud Data Engineers - ML

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



CLOUD DATA ENGINEER

- Experience in ETL/ELT
- Coding skills in Python or Java
- Knowledge of database query languages such as SQL
- Experience with Streaming
- Knowledge of Cloud Tools



CAT

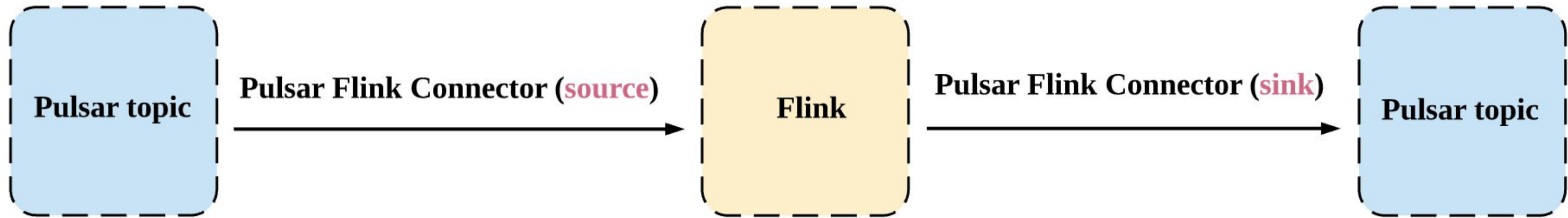
- Expert in ETL (Eating, Ties and Laziness)
- Edge Camera Interaction
- Typical User
- No Coding Skills
- Can use NiFi
- Questions your cloud spend



AI / Deep Learning / ML / DS

- Can run in Apache NiFi
- Can run in Apache Pulsar Functions
- Can run in Apache Flink
- Can run in Apache NiFi - MiNiFi Agents

FLiP Stack (FLink -integrate- Pulsar)



<https://hub.streamnative.io/data-processing/pulsar-flink/2.7.0/>



WHAT IS APACHE NIFI?

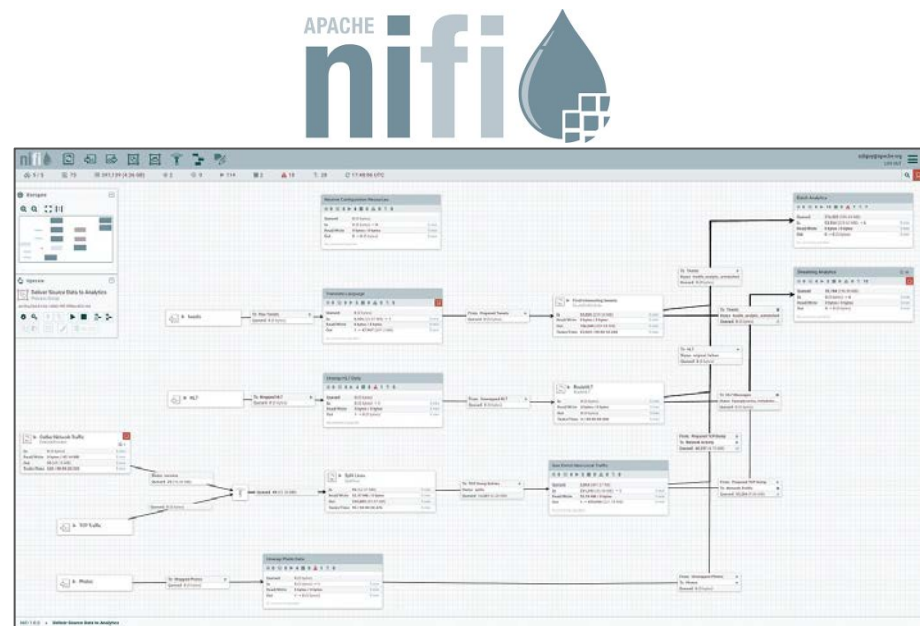
Apache NiFi is a scalable, real-time streaming data platform that collects, curates, and analyzes data so customers gain key insights for immediate actionable intelligence.



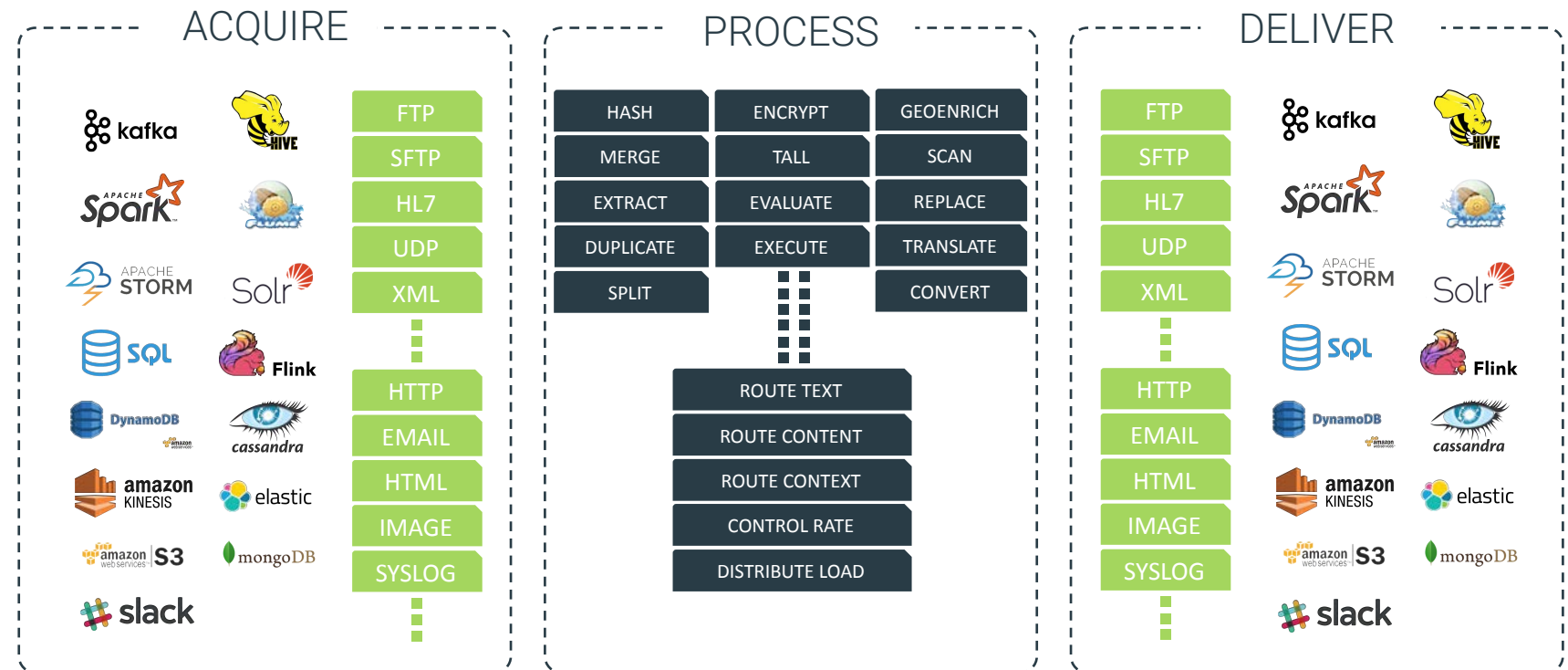


APACHE NIFI

Enable easy ingestion, routing, management and delivery of any data anywhere (Edge, cloud, data center) to any downstream system with built in end-to-end security and provenance



Advanced tooling to industrialize flow development (Flow Development Life Cycle)



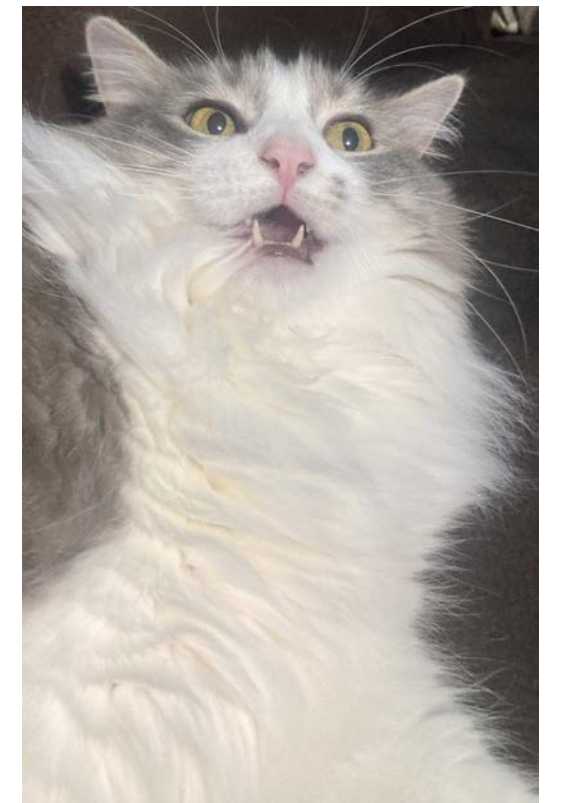
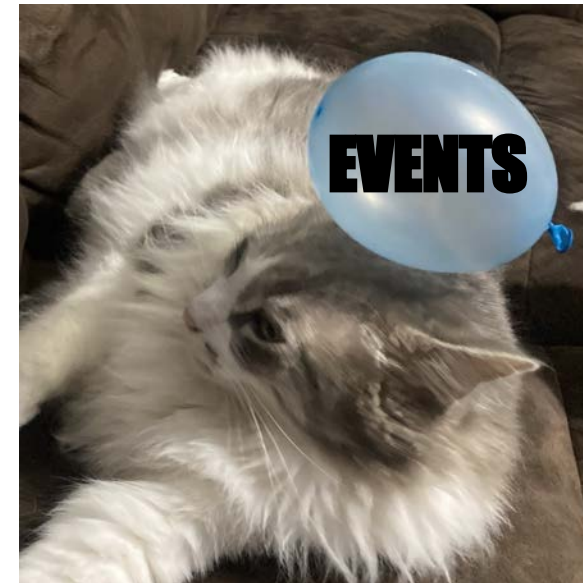
- Over 300 Prebuilt Processors
- Easy to build your own
- Parse, Enrich & Apply Schema
- Filter, Split, Merger & Route
- Throttle & Backpressure

- Guaranteed Delivery
- Full data provenance from acquisition to delivery
- Diverse, Non-Traditional Sources
- Eco-system integration

WHAT IS APACHE PULSAR?



Apache Pulsar is an open source, cloud-native distributed messaging and streaming platform.

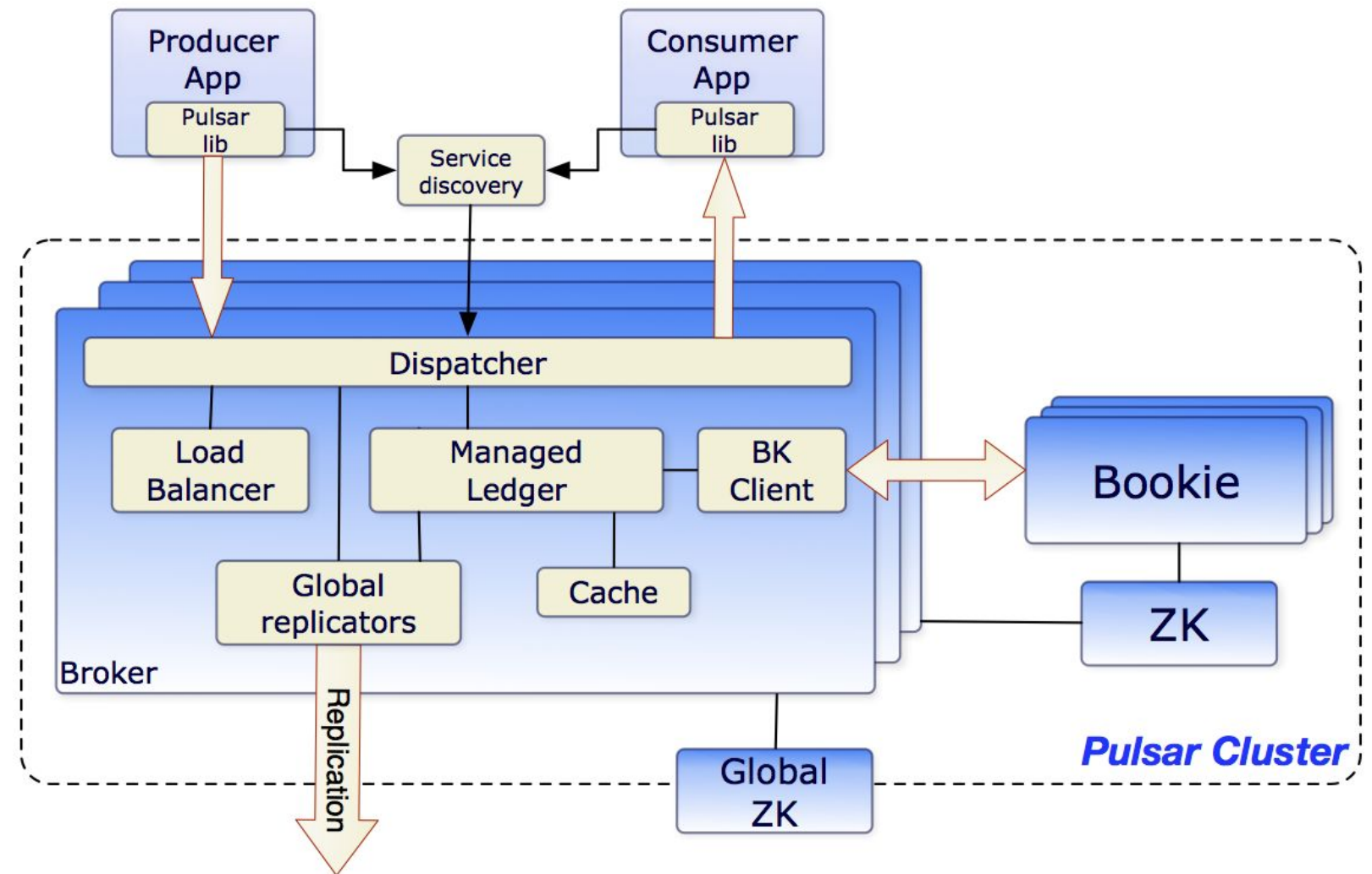




APACHE PULSAR

Enable Geo-Replicated Messaging

- Pub-Sub
- Geo-Replication
- Pulsar Functions
- Horizontal Scalability
- Multi-tenancy
- Tiered Persistent Storage
- Pulsar Connectors
- REST API
- CLI
- Many clients available
- Four Different Subscription Types
- Multi-Protocol Support
 - MQTT
 - AMQP
 - JMS
 - Kafka
 - ...



APACHE FLINK

3B+

3B+ data points daily streaming in from 25 million customers running real time machine learning prediction



Flink

USE CASE

Streaming real-time data pipelines that need to handle complex stream or batch data event processing, analytics, and/or support event-driven applications

TECHNOLOGY

Flink performs compute at in-memory speed at any scale
Flink parses SQL using Apache Calcite, which supports standard ANSI SQL
Flink runs standalone, on YARN, and has a K8s Operator

APPLICATION

Comcast a global media uses Flink for operationalizing machine learning models and near-real-time event stream processing
Flink helps deliver a personalized, contextual interaction reducing time to support resolutions saving millions of dollars per year

CONSIDERATION

Data Freshness SLAs
Flink can read and write from Hive data
Review requirements for fault tolerance, resilience, and HA

Apache MXNet Native Processor through DJL.AI for Apache NiFi



```
#workshop
11:30 AM =====
Deep Learning Class Label: person
File: cc0a469f-c108-42c7-95c6-10e5fda95006.person.png
Probability: 0.96
UUID: 32ef65a3-0650-42cd-965c-ba25597eb1ad
Rank: 1
Bounding Box (Height/Width, X,Y)
0.74/ 0.69
0.27, 0.25
Image (Height/Width, X,Y)
480 / 640
0, 0
=====
```

tspann 11:30 AM
371bdb8f-35bc-4a2a-919c-bdeb609b726c.person.png



```
return Paths.get(uri).toAbsolutePath().getParent().toString();
}

private void runAndAssertHappy() {
    testRunner.setValidateExpressionUsage(false);
    testRunner.run();
    testRunner.assertValid();

    testRunner.assertAllFlowFilesTransferred(DeepLearningProcessor.REL_SUCCESS);
    List<MockFlowFile> successFiles = testRunner.getFlowFilesForRelationship(DeepLearningProcessor.REL_SUCCESS);

    for (MockFlowFile mockFile : successFiles) {
        assertEquals("mocked: " + mockFile.getAttributes().get("class_1"), mockFile.getAttributes().get("probability_1"));
        System.out.println("Size: " + mockFile.getSize());
        Map<String, String> attributes = mockFile.getAttributes();
        for (String attribute : attributes.keySet()) {
            System.out.println("Attribute: " + attribute);
        }
    }
}

@Test
public void testProcessor() throws Exception {
    java.io.File resourcesDirectory = new java.io.File(
        System.out.println(resourcesDirectory.getAbsolutePath());
    testRunner.setProperty(DeepLearningProcessor.BACKBONE, "mxnet");
    testRunner.setProperty(DeepLearningProcessor.DATASET, "mnist");
    testRunner.run();
}

Run: DeepLearningProcessorTest.testProcessor
Tests passed: 1 of 1 test - 4 s 618 ms
DeepLearningProcessorTest (com 4 s 818 ms)
testProcessor 4 s 818 ms
Size:1761238
Attribute:boundingbox_height_1 = 0.15
Attribute:probability_1 = 1.00
Attribute:image_min_y_1 = 0
Attribute:image_min_x_1 = 0
Attribute:rank_1 = 1
Attribute:uuid = e5093c52-15ab-4849-8876-a25796714904
Attribute:boundingbox_width_1 = 0.24
```

Attribute Values

- boundingbox_height_1: 0.99
- No value set
- boundingbox_width_1: 0.90
- No value set
- boundingbox_x_1: 0.09
- No value set
- boundingbox_y_1: 0.01
- No value set
- class_1: tvmonitor
- No value set
- filename: 2020-08-26_1330.jpg.tvmonitor.png
- 2020-08-26_1330.jpg (previous)

This processor uses the DJL.AI Java Interface

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

<https://dev.to/tspannhw/easy-deep-learning-in-apache-nifi-with-djl-2d79>

Apache MXNet Native Processor for Apache NiFi

- <https://www.slideshare.net/bunkertor/apache-deep-learning-101-apachecon-montreal-2018-v031>
- <https://www.slideshare.net/bunkertor/apache-deep-learning-202-washington-dc-dws-2019>
- <https://www.slideshare.net/bunkertor/apache-deep-learning-201-barcelona-dws-march-2019>



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!



FlowFile

DETAILS ATTRIBUTES

Attribute Values

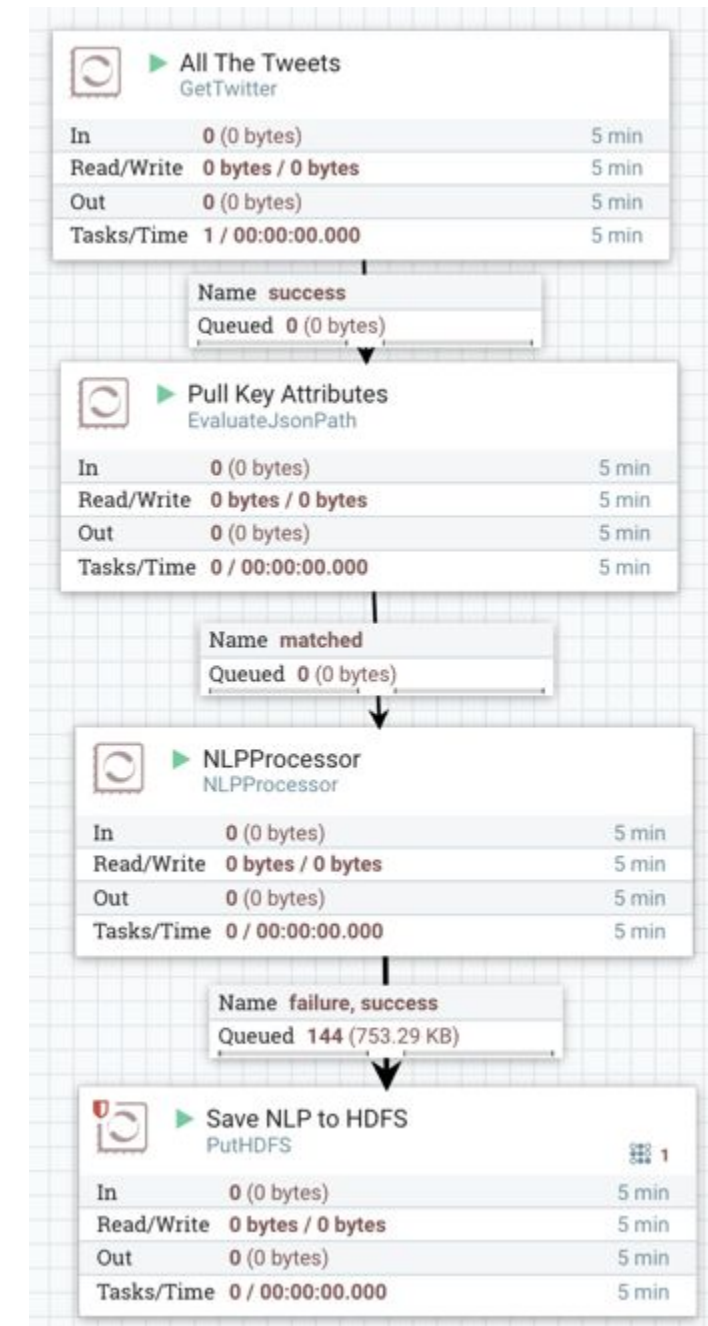
filename
2788601463132800.json

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

followers_count
47

location
Columbus, Ohio

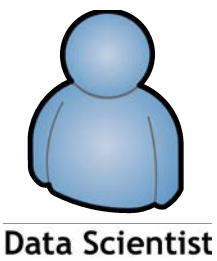
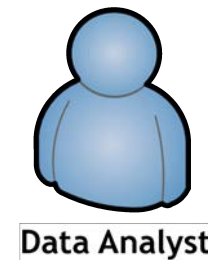
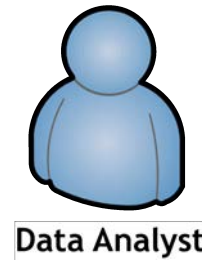
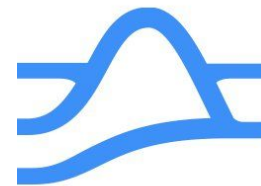
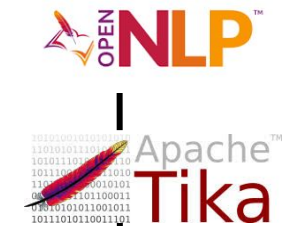
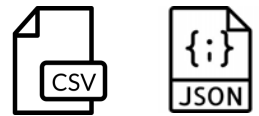
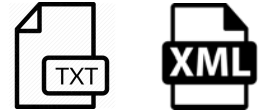
locations
{"locations":[{"location":"Sydney"}]}



<https://community.cloudera.com/t5/Community-Articles/Open-NLP-Example-Apache-NiFi-Processor/ta-p/249293>

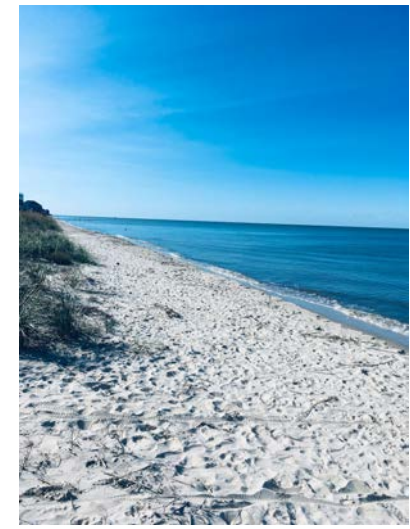
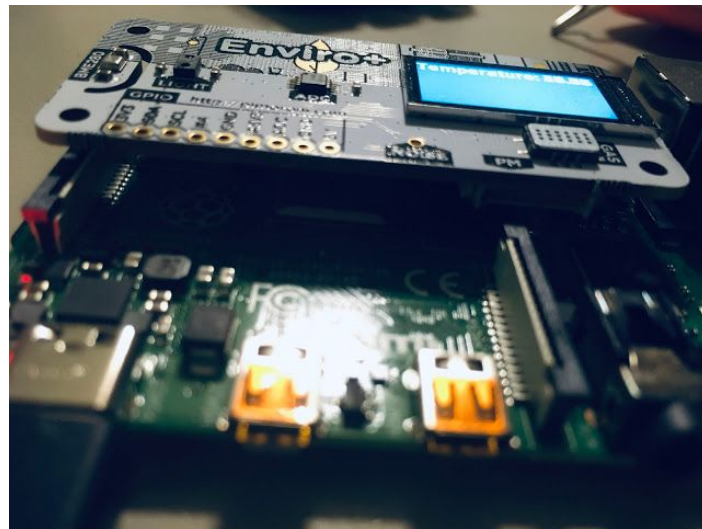
<https://opennlp.apache.org/news/release-190.html>

ALL DATA - ANYTIME - ANYWHERE - ANY CLOUD



SHOW ME SOME DATA

```
{"uuid": "rpi4_uuid_jfx_20200826203733", "amplitude100": 1.2, "amplitude500": 0.6, "amplitude1000": 0.3, "lownoise": 0.6, "midnoise": 0.2, "highnoise": 0.2, "amps": 0.3, "ipaddress": "192.168.1.76", "host": "rp4", "host_name": "rp4", "macaddress": "6e:37:12:08:63:e1", "systemtime": "08/26/2020 16:37:34", "endtime": "1598474254.75", "runtime": "28179.03", "starttime": "08/26/2020 08:47:54", "cpu": 48.3, "cpu_temp": "72.0", "diskusage": "40219.3 MB", "memory": 24.3, "id": "20200826203733_28ce9520-6832-4f80-b17d-f36c21fd8fc9", "temperature": "47.2", "adjtemp": "35.8", "adjtempf": "76.4", "temperaturef": "97.0", "pressure": 1010.0, "humidity": 8.3, "lux": 67.4, "proximity": 0, "oxidising": 77.9, "reducing": 184.6, "nh3": 144.7, "gasKO": "Oxidising: 77913.04 Ohms\nReducing: 184625.00 Ohms\nNH3: 144651.47 Ohms"}
```

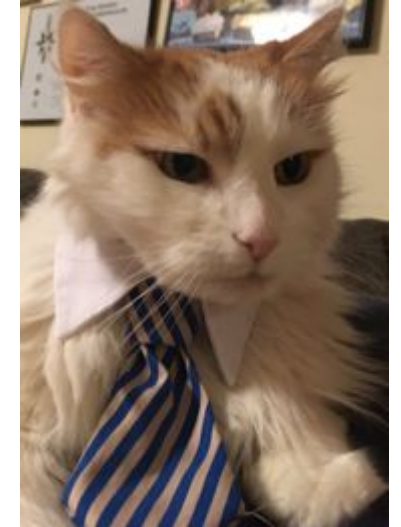


Weather Streaming Pipeline

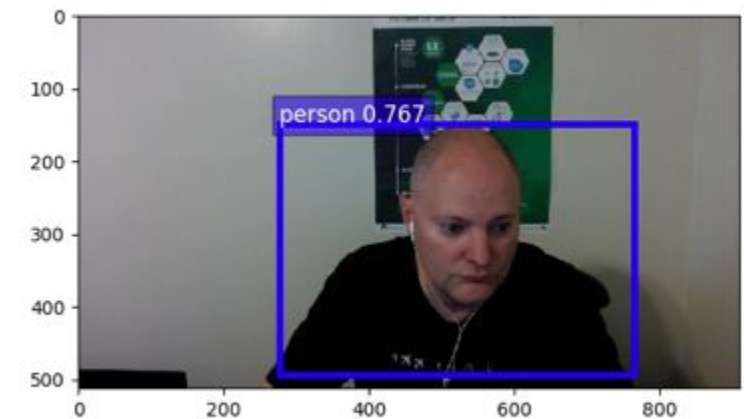
Weather

location	observation_time	credit	credit_url	image
Abingdon, VA	Last Updated on Oct 27 2020, 11:55 am EDT	NOAA's National Weather Service	http://weather.gov/	MapRecord[link=http://weather.gov, title=NOAA's Nati
Ada, Ada Municipal Airport, OK	Last Updated on Oct 27 2020, 10:55 am CDT	NOAA's National Weather Service	http://weather.gov/	MapRecord[link=http://weather.gov, title=NOAA's Nati
Adrian, Lenawee County Airport, MI	Last Updated on Oct 27 2020, 11:53 am EDT	NOAA's National Weather Service	http://weather.gov/	MapRecord[link=http://weather.gov, title=NOAA's Nati
Adrian, Lenawee County Airport, MI	Last Updated on Oct 27 2020, 12:53 pm EDT	NOAA's National Weather Service	http://weather.gov/	MapRecord[link=http://weather.gov, title=NOAA's Nati
Afton WY, WY	Last Updated on Oct 27 2020, 9:55 am MDT	NOAA's National Weather Service	http://weather.gov/	MapRecord[link=http://weather.gov, title=NOAA's Nati
Aiken Municipal Airport, SC	Last Updated on Oct 27 2020, 11:55 am EDT	NOAA's National Weather Service	http://weather.gov/	MapRecord[link=http://weather.gov, title=NOAA's Nati
Ak-Chin Regional Airport, AZ	Last Updated on Oct 27 2020, 9:55 am MST	NOAA's National Weather Service	http://weather.gov/	MapRecord[link=http://weather.gov, title=NOAA's Nati
Akron Canton Regional Airport, OH	Last Updated on Oct 27 2020, 12:51 pm EDT	NOAA's National Weather Service	http://weather.gov/	MapRecord[link=http://weather.gov, title=NOAA's Nati
Alabaster, Shelby County Airport, AL	Last Updated on Oct 27 2020, 10:53 am CDT	NOAA's National Weather Service	http://weather.gov/	MapRecord[link=http://weather.gov, title=NOAA's Nati
Albert Whitted Airport, FL	Last Updated on Oct 27 2020, 12:53 pm EDT	NOAA's National Weather Service	http://weather.gov/	MapRecord[link=http://weather.gov, title=NOAA's Nati
Albuquerque, Double Eagle II Airport, NM	Last Updated on Oct 27 2020, 10:55 am MDT	NOAA's National Weather Service	http://weather.gov/	MapRecord[link=http://weather.gov, title=NOAA's Nati

< 1 2 3 4 5 >



weather map





DEEPER CONTENT

- <https://www.datainmotion.dev/2020/10/running-flink-sql-against-kafka-using.html>
- <https://www.datainmotion.dev/2020/10/top-25-use-cases-of-cloudera-flow.html>
- <https://github.com/tspannhw/EverythingApacheNiFi>
- <https://github.com/tspannhw/CloudDemo2021>
- <https://github.com/tspannhw/StreamingSQLExamples>
- <https://www.linkedin.com/pulse/2021-schedule-tim-spann/>
- <https://github.com/tspannhw/StreamingSQLExamples/blob/8d02e62260e82b027b43abb911b5c366a3081927/README.md>



THANK YOU

