Streaming IoT sensor data with LocationTech GeoMesa, Apache Kafka, and NiFi
LocationTech
GeoMesa
Overview
What is GeoMesa?

A suite of tools for streaming, persisting, managing, and analyzing spatio-temporal data at scale
What is GeoMesa?

A suite of tools for **streaming**, **persisting**, managing, and analyzing spatio-temporal data at scale
What is GeoMesa?

A suite of tools for streaming, **persisting**, managing, and analyzing spatio-temporal data at scale.
What is GeoMesa?

A suite of tools for streaming, persisting, managing, and analyzing spatio-temporal data at scale
What is GeoMesa?

A suite of tools for streaming, **persisting**, managing, and **analyzing** spatio-temporal data at scale
Proposed Reference Architecture
Data Sources

Maritime - AIS

ORBCOMM

Air domain - ADS-B

ADS-B Exchange

Contextual

Weather
Administrative Regions
Basemaps

Optix Cloud

Ingest and Data Management

Real-time Streaming Data

Big Data Persistence

Big Data Analytics

Big Data Visualization

STEALTH

OGC web-tier

GeoServer

ADS-B Exchange Data Sources

Maritime - AIS

Air domain - ADS-B

Contextual

Weather
Administrative Regions
Basemaps

Optix Cloud

Ingest and Data Management

Real-time Streaming Data

Big Data Persistence

Big Data Analytics

Big Data Visualization

STEALTH

OGC web-tier

GeoServer

GENERAL ATOMICS CRI
Demo!
What did we see in the demo?

Live Data:

- Live View of all maritime vessels
- Activities layer
  - Port Arrivals / Departures
  - Status changes
  - Etc…

Historical Data:

- Track History

Contextual Data:

- Basemap
- Ports
- EEZs, etc...
What technologies enable the demo?

**Live Data:**
- Live View of all maritime vessels

**Activities layer**
- Port Arrivals / Departures
- Status changes, Etc...

**Historical Data:**
- Track History

**Contextual Data:**
- Basemap
- Ports, EEZs, etc...

**Live Data:**
- GeoMesa’s Kafka DataStore

**Activities:**
- Siddhi / KStreams, etc

**Historical / Contextual Data:**
- GeoMesa HBase + NiFi

[General Atomics CRI logo]
Kafka

- Kafka as a Message Bus
- In-memory table view
What is Kafka?

- Kafka is a distributed Write-Ahead Log.
  - **WAL =>**
    - Writes are fast
    - Reads can be fast since they can be batched
  - Distributed
    - Can turn up the parallelism
How do we get moving DOTM?

Producers get data from somewhere

- They create SimpleFeatures
- Write them to a Kafka topic
- … in What format?
  - Usually Kryo
  - Optionally you can switch to Avro

What do the (Kryo) messages look like?

- CreateOrUpdate(id)
- Delete(id)
- Clear()
What are all those settings?

- Event-time lets one set Kafka expiration based on an attribute
- Log compaction reduces the size of the Kafka WAL
GeoMesa Kafka DataStore In-Memory Database

GeoMesa KDS clients (like GeoServer)

- Listen for updates from Kafka
- Receive and answer spatial queries

These clients need an in-memory database structure that can be updated quickly as new updates come in.
GeoMesa Kafka DataStore In-Memory Database

GeoMesa KDS clients (like GeoServer)

- Listen for updates from Kafka
- Receive and answer spatial queries

These clients need an in-memory database structure that can be updated quickly as new updates come in.

Usually spatial data structures like R-Trees and Quad-trees would be slower to update in light of the volume of updates.

Other possibilities include trying H2’s spatial support. Indexing in H2 was slow when we tried it. (Admittedly, back in 2016.)

To address this, GeoMesa has rolled its own lightweight, in-memory database.
For most use cases, GeoMesa uses a class which maintains two things:

1. A HashMap of Feature IDs to records
2. A bucket index of spatial grid cells containing records

Updates:

- Find the old record in the HashMap
- Remove it from the bucket index
- Add the new element
GeoMesa Kafka DataStore In-Memory Database

For situations when queries on attribute columns may be important, GeoMesa can be configured to use CQEngine!

For GeoServer use cases, it is faster than the standard KDS and H2.

GeoMesa Kafka DataStore In-Memory Database

![Graph showing query time for different algorithms and data types.]

- **Attribute-only**
  - Orig. (geo)
  - CQE (geo)
  - CQE (all)
  - H2 (geo)
  - H2 (all)

- **Spatial-only**
  - Orig. (geo)
  - CQE (geo)
  - CQE (all)
  - H2 (geo)
  - H2 (all)

- **Attribute and spatial**
  - Orig. (geo)
  - CQE (geo)
  - CQE (all)
  - H2 (geo)
  - H2 (all)
What tools are there?

Kafka has command line tools

- Manage topics
- Send messages
- Listen to topics

GeoMesa Kafka has command line tools

- Manage SimpleFeatureTypes
- Send SimpleFeatures as messages
- Listen to topics
What is NiFi?

From https://nifi.apache.org/

Apache NiFi supports powerful and scalable directed graphs of data routing, transformation, and system mediation logic. Some of the high-level capabilities and objectives of Apache NiFi include:

- **Web-based user interface**
  - Seamless experience between design, control, feedback, and monitoring

- **Highly configurable**
  - Flow can be modified at runtime
  - Back pressure

- **Data Provenance**
  - Track dataflow from beginning to end

- **Designed for extension**
  - Build your own processors and more
  - Enables rapid development and effective testing
What is NiFi?

From https://nifi.apache.org/

Apache NiFi supports powerful and scalable directed graphs of data routing, transformation, and system mediation logic. Some of the high-level capabilities and objectives of Apache NiFi include:

- **Web-based user interface**
  - Seamless experience between design, control, feedback, and monitoring

- **Highly configurable**
  - Flow can be modified at runtime
  - Back pressure

- **Data Provenance**
  - Track dataflow from beginning to end

- **Designed for extension**
  - Build your own processors and more
  - Enables rapid development and effective testing
How do we use NiFi?

We typically use NiFi for

- Managing data flows
- ETL
  - Extract
  - Transform
  - Load
How do we use NiFi?

We typically use NiFi for

- Managing data flows
- ETL
  - Extract
  - Transform
  - Load

As an example, one could:

Use a GetHTTP or ListenTCP processor to extract data from a source.

A processor like TransformRecord or TransformXML can be used to transform data in flow files.

Processors like PutJDBC, PutTCP, or PutS3 can load data into external systems.
How do we use NiFi?

We typically use NiFi for

- Managing data flows
- ETL
  - Extract
  - Transform
  - Load

As an example, one could:

Use a GetHTTP or ListenTCP processor to extract data from a source.

A processor like TransformRecord or TransformXML can be used to transform data in flow files.

Processors like PutJDBC, PutTCP, or PutS3 can load data into external systems.

Let’s do this for SimpleFeatures and GeoMesa!
GeoMesa-NiFi

GeoMesa-NiFi is a GeoMesa community project to add NiFi processors and components to help NiFi users integrate GeoMesa into their NiFi flows.

### Extract: Reads SimpleFeatures from a GeoMesa managed Kafka topic.

**GetGeoMesaKafkaRecord**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetGeoMesaKafkaRecord</td>
<td>Get GeoMesa Kafka Record</td>
</tr>
</tbody>
</table>

| In | 0 (0 bytes) | 5 min |
| Read/Write | 0 bytes / 0 bytes | 5 min |
| Out | 0 (0 bytes) | 5 min |
| Tasks/Time | 0 / 00:00:00.000 | 5 min |

### Transform: Applies a GeoMesa converter to create GeoAvro files.

**ConvertToGeoAvro**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvertToGeoAvro</td>
<td>Convert GeoMesa to GeoAvro</td>
</tr>
</tbody>
</table>

| In | 0 (0 bytes) | 5 min |
| Read/Write | 0 bytes / 0 bytes | 5 min |
| Out | 0 (0 bytes) | 5 min |
| Tasks/Time | 0 / 00:00:00.000 | 5 min |
GeoMesa-NiFi

GeoMesa-NiFi is a GeoMesa community project to add NiFi processors and components to help NiFi users integrate GeoMesa into their NiFi flows.
GeoMesa-NiFi

GeoMesa-NiFi is a GeoMesa community project to add NiFi processors and components to help NiFi users integrate GeoMesa into their NiFi flows.
GeoMesa-NiFi

In addition to processors, there are

- **Configuration Services**
  - To manage the configuration for connecting to datastores
- **GeoAvroRecordSetWriter**
  - Allows any processor which write out NiFi RecordSets to create GeoAvro files.
Put it all together in a GeoMesa use case

One can build a flow that:

- Reads from a source (like TCP)
- Use `ConvertToAvro` to process the raw data into SimpleFeatures and write it out as Avro
- The Avro can be saved in S3 with `PutS3`
- The Avro files can be loaded into HBase with `PutGeoMesaHBase`
- The Avro files can also be sent to Kafka with `PutGeoMesaKafka`
Streaming Analytics
What is GeoMesa?

A suite of tools for **streaming**, **persisting**, managing, and analyzing spatio-temporal data at scale
ksqlDB Geospatial Integration

https://github.com/wlaforest/KSQLGeo

Add some spatial UDFs to ksqlDB

https://github.com/wlaforest/KafkaGeoDemo

Demo of the UDFs
Thanks!

- jhughes@ccri.com
- https://www.geomesa.org/
- https://gitter.im/locationtech/geomesa
- https://github.com/locationtech/geomesa
- Twitter @CCR_inc

CCRi is hiring!

https://www.ccri.com/careers/

- DevOps
- Software Engineers
- Data Scientists