State of LocationTech GeoMesa

FOSS4G 2021 Buenos Aires
Jim Hughes
September 30, 2021

GENERAL ATOMICS
CCRi

LocationTech GeoMesa Overview

GENERAL ATOMICS
CCRi

What is GeoMesa?

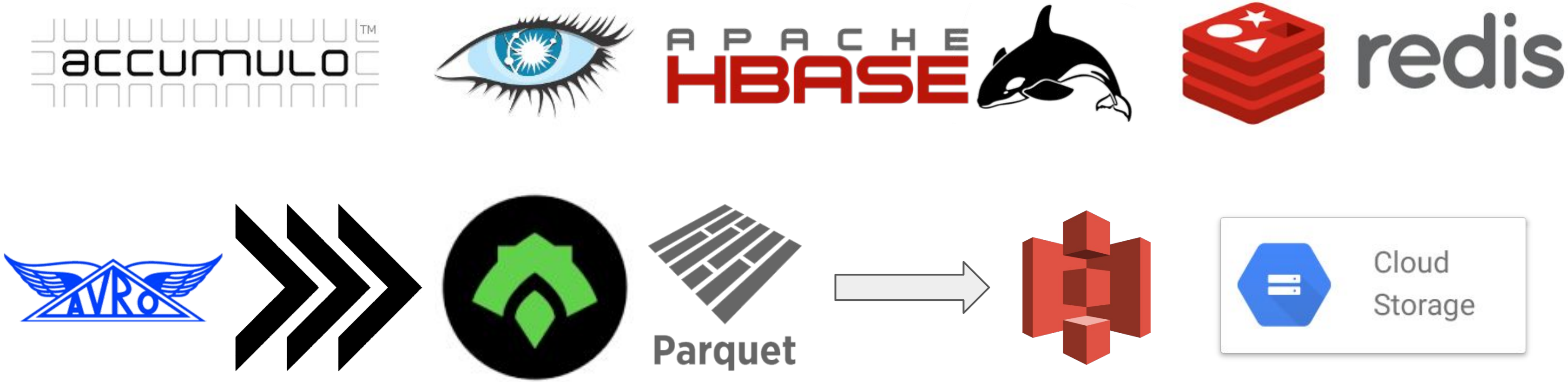
A suite of tools for streaming, persisting, managing, and analyzing spatio-temporal data at scale

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

APACHE nifi

minifi

What is GeoMesa?

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

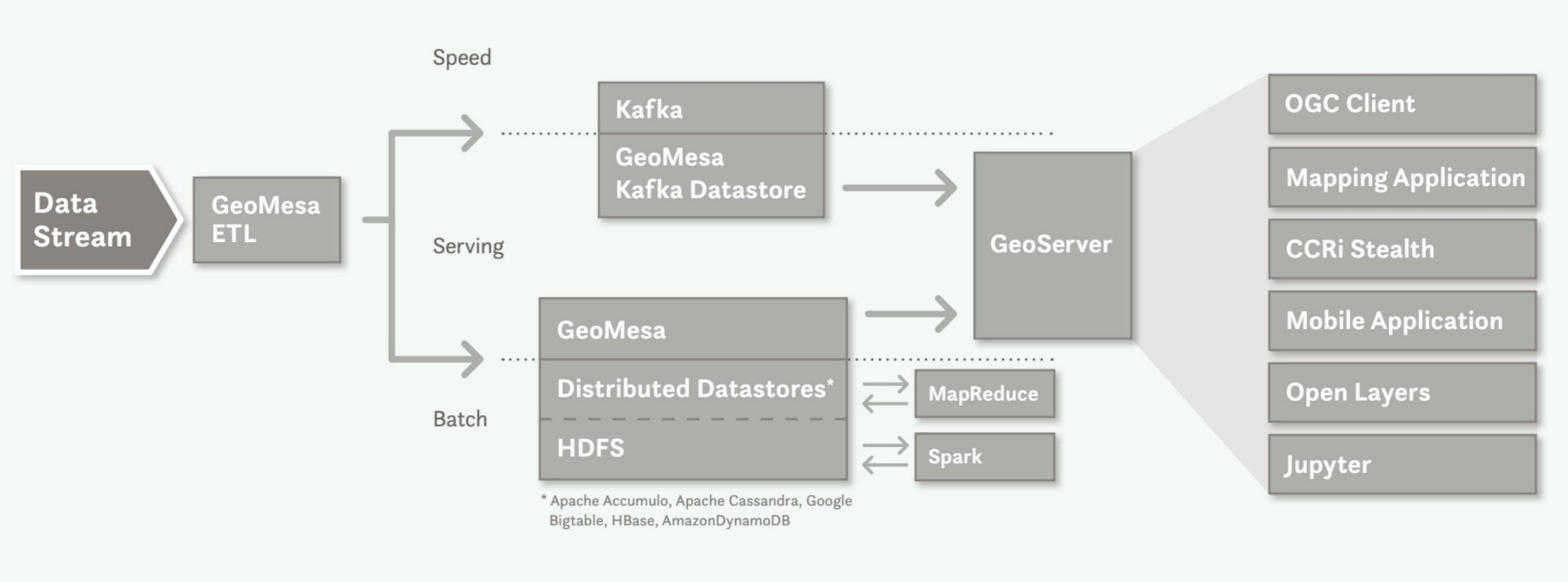
APACHE Spark™

hadoop

jupyter

Apache Zeppelin

Proposed Reference Architecture



* Apache Accumulo, Apache Cassandra, Google Bigtable, HBase, AmazonDynamoDB

GeoMesa 3.0

Released July 2020

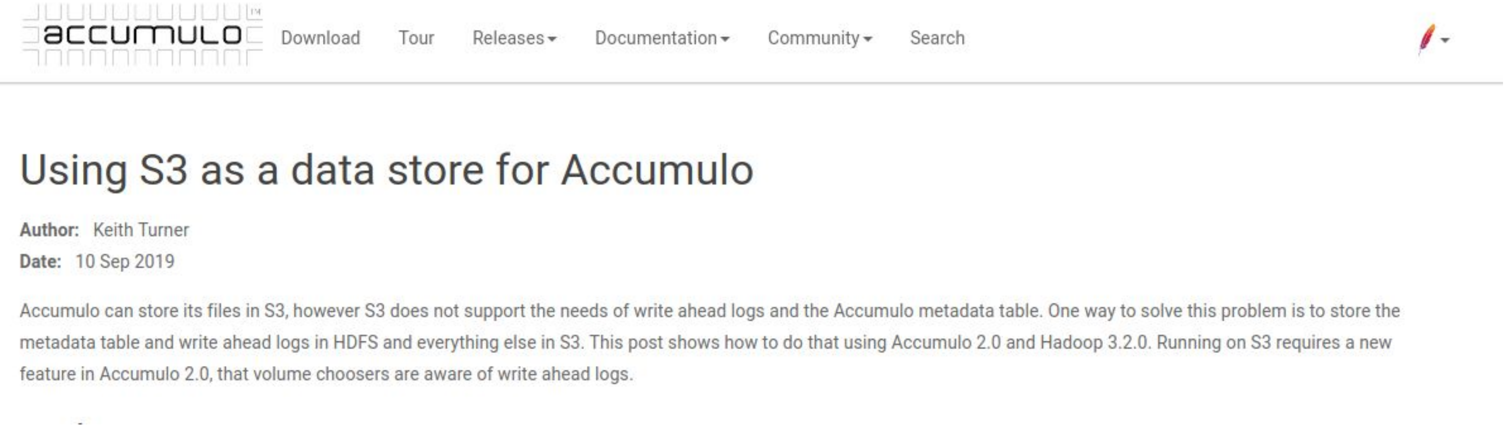
Persistence / Databases:

- Accumulo 2.0 support with S3!

Accumulo 2.0 support with S3!

Accumulo 2.0 adds support for AWS S3.

- This allows for much cheaper cloud storage.
- Backend storage can be reduced by 90%+ in some cases.



Using S3 as a data store for Accumulo

Author: Keith Turner
Date: 10 Sep 2019

Accumulo can store its files in S3, however S3 does not support the needs of write ahead logs and the Accumulo metadata table. One way to solve this problem is to store the metadata table and write ahead logs in HDFS and everything else in S3. This post shows how to do that using Accumulo 2.0 and Hadoop 3.2.0. Running on S3 requires a new feature in Accumulo 2.0, that volume choosers are aware of write ahead logs.

GeoMesa 3.1

Released October 2020

GEneral ATOMICS
CCRi

GeoMesa NiFi

- What is Apache NiFi?
- GetGeoMesaKafka Record
- GeoAvro RecordSetWriter

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** **<- GeoMesa-NiFi Processors**
  - **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.

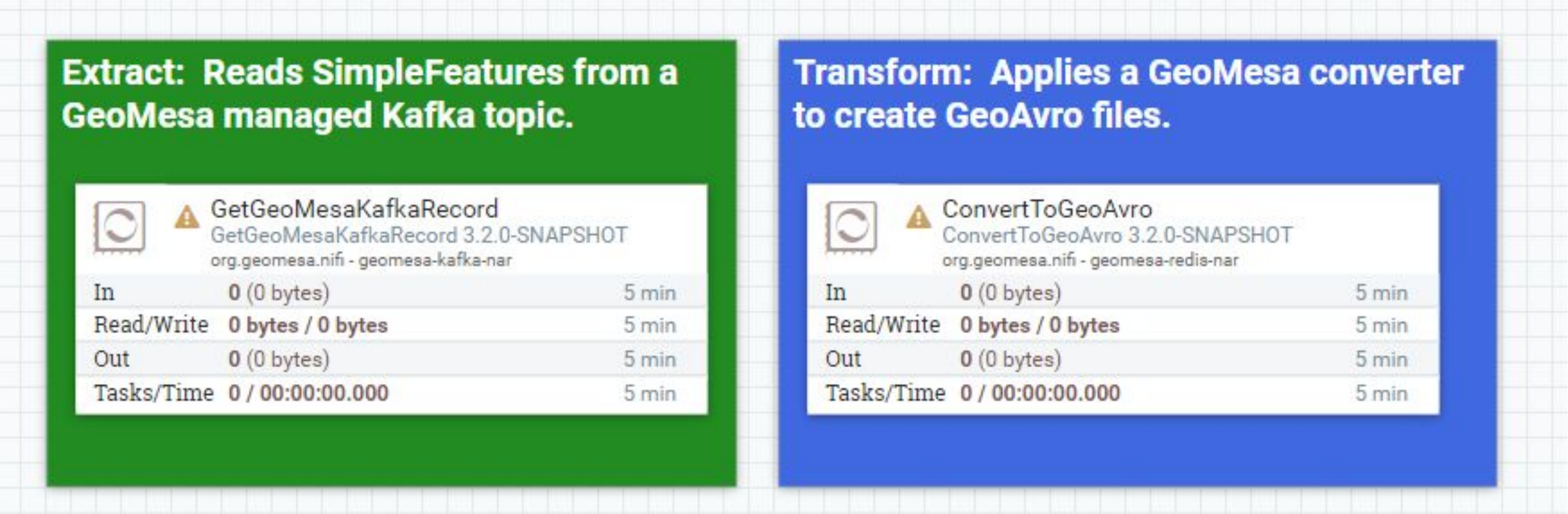
A processor like **TransformRecord** or **TransformXML** can be used to *transform* data in flow files.

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
GetGeoMesaKafkaRecord 3.2.0-SNAPSHOT
org.geomesa.nifi - geomesa-kafka-nar

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

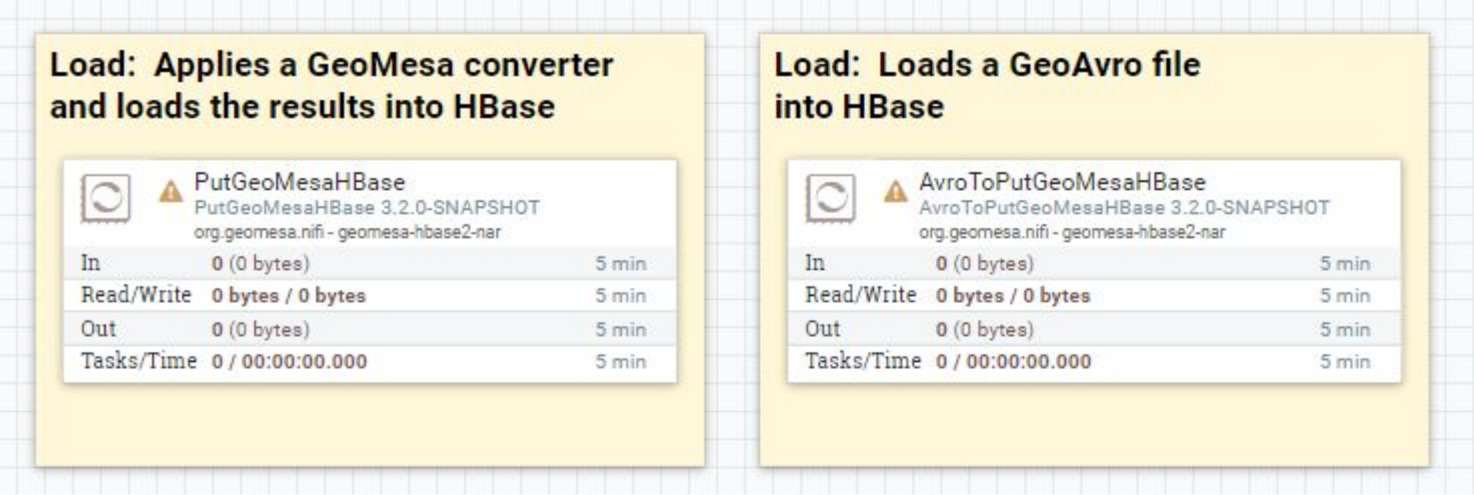
Transform: Applies a GeoMesa converter to create GeoAvro files.

ConvertToGeoAvro
ConvertToGeoAvro 3.2.0-SNAPSHOT
org.geomesa.nifi - geomesa-redis-nar

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.



Load: Applies a GeoMesa converter and loads the results into HBase

PutGeoMesaHBase
PutGeoMesaHBase 3.2.0-SNAPSHOT
org.geomesa.nifi - geomesa-hbase2-nar

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

Load: Loads a GeoAvro file into HBase

AvroToPutGeoMesaHBase
AvroToPutGeoMesaHBase 3.2.0-SNAPSHOT
org.geomesa.nifi - geomesa-hbase2-nar

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

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.

GeoMesa 3.2

Released April 2021

GeoMesa-NiFi

- Support for modifying writes

Spark

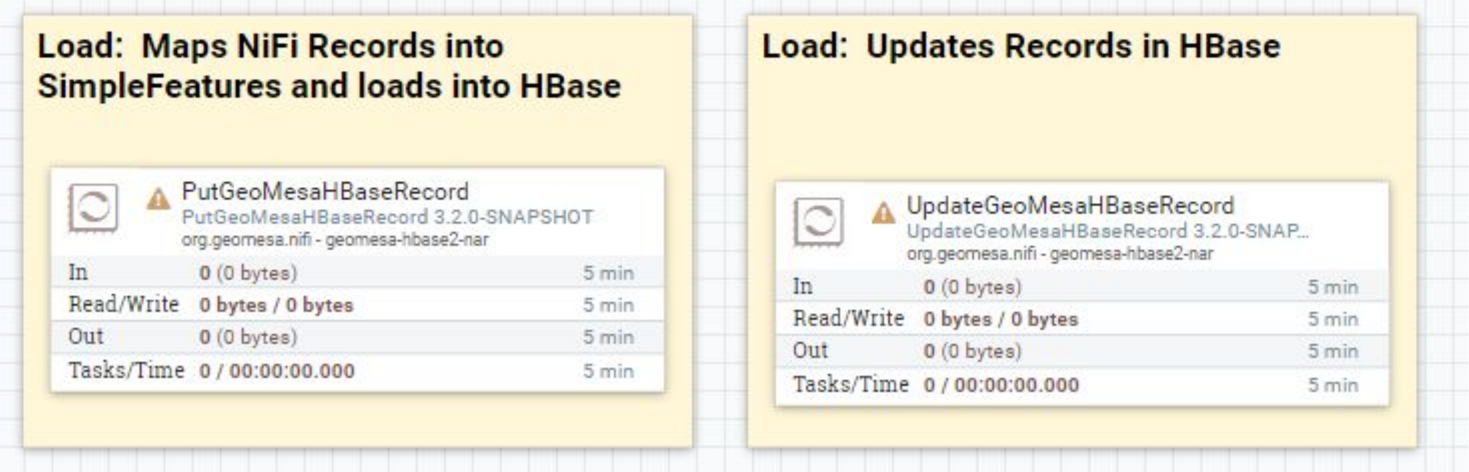
- Added Scala 2.12 support
- Added Spark 3.0/3.1.1 support

Streaming

- Transactional Writes
- Metrics Integration
- Readiness Check (GM-GS)

NiFi: Support for modifying writes

UpdateGeoMesa*Record Processors are new in 3.2.0!



Load: Maps NiFi Records into SimpleFeatures and loads into HBase

PutGeoMesaHBaseRecord
PutGeoMesaHBaseRecord 3.2.0-SNAPSHOT
org.geomesa.nifi - geomesa-hbase2-nar

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

Load: Updates Records in HBase

UpdateGeoMesaHBaseRecord
UpdateGeoMesaHBaseRecord 3.2.0-SNAP...
org.geomesa.nifi - geomesa-hbase2-nar

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

Spark: Added Spark 3.0/3.1.1 support

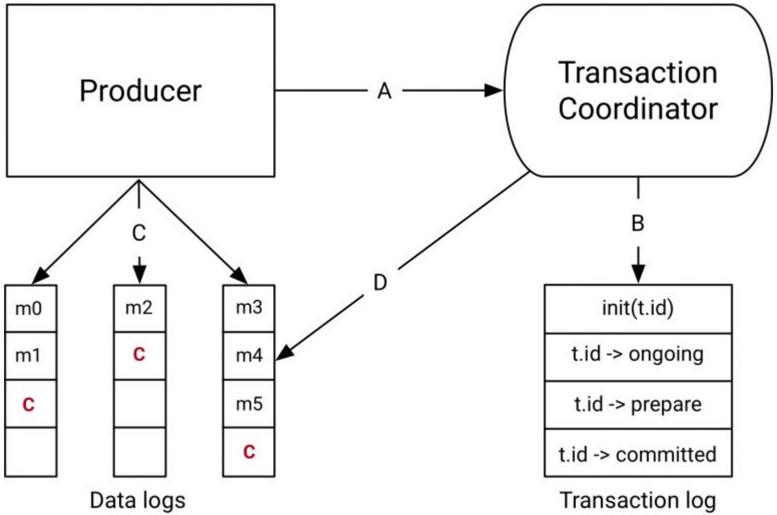
Required adding Scala 2.12 version of the build.

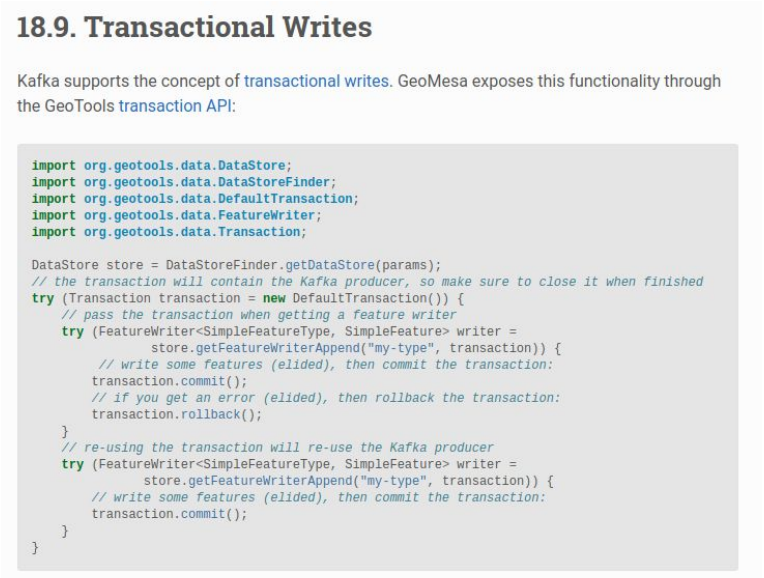
GeoMesa Version	Supported Scala Versions	Supported Spark Versions
2.x	2.11	Up to 2.4.x
3.0 - 3.1.x	2.11	Up to 2.4.x
3.2+	2.11 support deprecated 2.12 support added	2.3/2.4 support deprecated 3.0 / 3.1+ support added
4.0	2.12	3.0+

Kafka: Transactional Writes

From https://www.geomesa.org/documentation/stable/user/kafka/transactional_writes.html

From https://www.confluent.io/blog/transactions-apache-kafka/





18.9. Transactional Writes

Kafka supports the concept of transactional writes. GeoMesa exposes this functionality through the GeoTools transaction API:

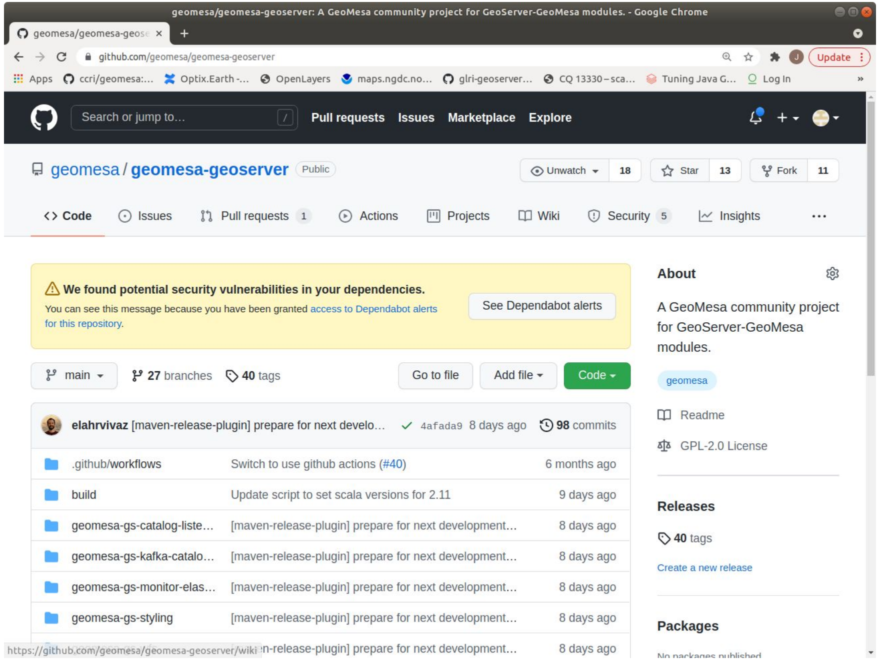
```java
import org.geotools.data.DataStore;
import org.geotools.data.DataStoreFinder;
import org.geotools.data.DefaultTransaction;
import org.geotools.data.FeatureWriter;
import org.geotools.data.Transaction;

DataStore store = DataStoreFinder.getDataStore(params);
// the transaction will contain the Kafka producer, so make sure to close it when finished
try (Transaction transaction = new DefaultTransaction()) {
    // pass the transaction when getting a feature writer
    try (FeatureWriter<SimpleFeatureType, SimpleFeature> writer =
                store.getFeatureWriterAppend("my-type", transaction)) {
         // write some features (elided), then commit the transaction:
        transaction.commit();
        // if you get an error (elided), then rollback the transaction:
        transaction.rollback();
    }
    // re-using the transaction will re-use the Kafka producer
    try (FeatureWriter<SimpleFeatureType, SimpleFeature> writer =
               store.getFeatureWriterAppend("my-type", transaction)) {
        // write some features (elided), then commit the transaction:
        transaction.commit();
    }
}
```

Kafka: Readiness Check (GeoMesa-GeoServer)

GeoMesa-GeoServer is a project to hold code that helps integrate extra GeoMesa capabilities into GeoServer.

GeoMesa implements GeoTools interfaces, so GeoMesa-GeoServer is not needed for all deployments.



geomesa/geomesa-geoserver: A GeoMesa community project for GeoServer-GeoMesa modules. - Google Chrome

github.com/geomesa/geomesa-geoserver

geomesa / geomesa-geoserver Public

⚠ We found potential security vulnerabilities in your dependencies.

You can see this message because you have been granted access to Dependabot alerts for this repository.

See Dependabot alerts

main · 27 branches · 40 tags

elahrvivaz [maven-release-plugin] prepare for next develo… ✓ 4afada9 8 days ago 98 commits

.github/workflows	Switch to use github actions (#40)	6 months ago
build	Update script to set scala versions for 2.11	9 days ago
geomesa-gs-catalog-liste…	[maven-release-plugin] prepare for next development…	8 days ago
geomesa-gs-kafka-catalo…	[maven-release-plugin] prepare for next development…	8 days ago
geomesa-gs-monitor-elas…	[maven-release-plugin] prepare for next development…	8 days ago
geomesa-gs-styling	[maven-release-plugin] prepare for next development…	8 days ago
	…n-release-plugin] prepare for next development…	8 days ago

https://github.com/geomesa/geomesa-geoserver/wiki

About

A GeoMesa community project for GeoServer-GeoMesa modules.

geomesa

Readme

GPL-2.0 License

Releases

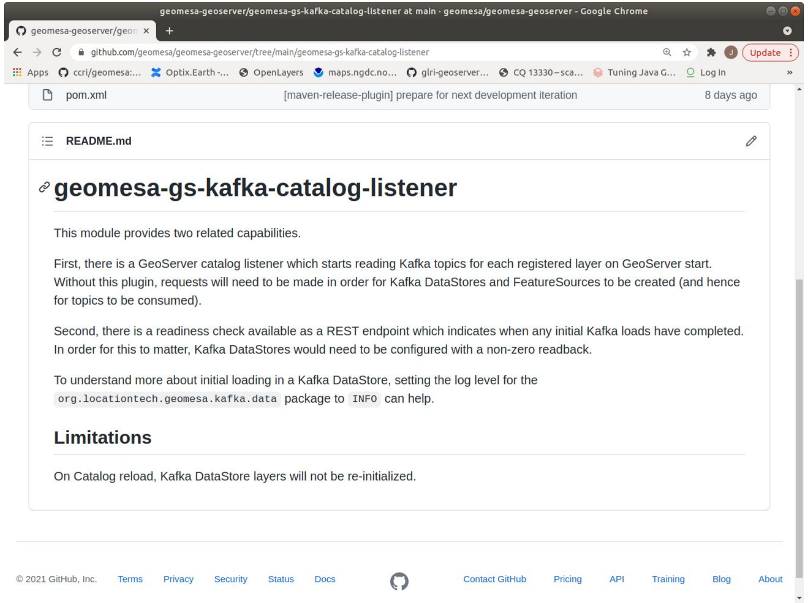
40 tags

Create a new release

Packages

No packages published

Kafka: Readiness Check (GeoMesa-GeoServer)



geomesa-gs-kafka-catalog-listener

This module provides two related capabilities.

First, there is a GeoServer catalog listener which starts reading Kafka topics for each registered layer on GeoServer start. Without this plugin, requests will need to be made in order for Kafka DataStores and FeatureSources to be created (and hence for topics to be consumed).

Second, there is a readiness check available as a REST endpoint which indicates when any initial Kafka loads have completed. In order for this to matter, Kafka DataStores would need to be configured with a non-zero readback.

To understand more about initial loading in a Kafka DataStore, setting the log level for the `org.locationtech.geomesa.kafka.data` package to `INFO` can help.

Limitations

On Catalog reload, Kafka DataStore layers will not be re-initialized.

The **geomesa-gs-kafka-catalog-listener** does two things:

- Starts reading from Kafka
- Provides an endpoint to show when all topics are loaded

This allows for GeoServer in a container environment (e.g. Kubernetes) to wait until it has all the data before being “ready”)

GeoMesa 3.3

Released September 2021

GeoMesa-NiFi

- PostGIS Write Support

Streaming

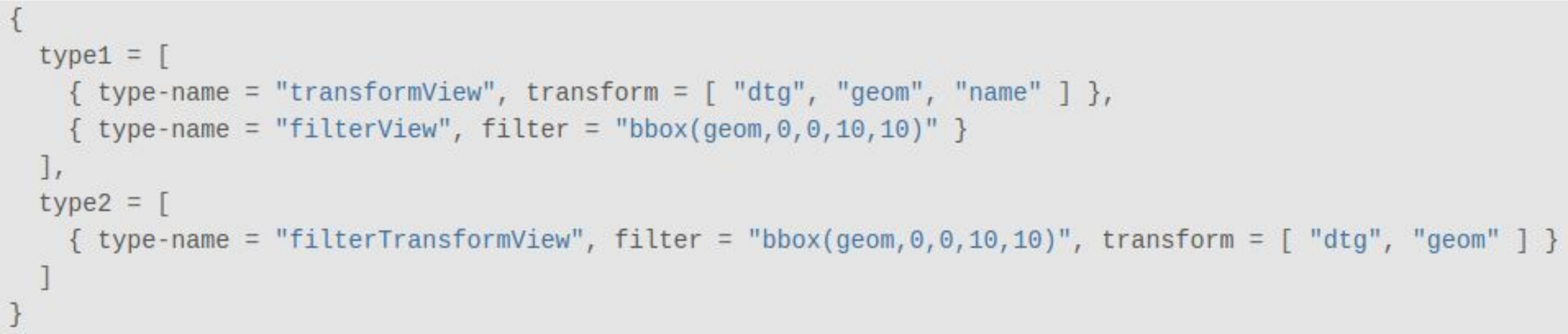
- Kafka Layer Views

Bulk Ingest into Accumulo

Improvements in FSDS Metadata

Kafka Layer Views

Expose your data in different ways to different users, without duplicating the storage and memory required.



```
{
  type1 = [
    { type-name = "transformView", transform = [ "dtg", "geom", "name" ] },
    { type-name = "filterView", filter = "bbox(geom,0,0,10,10)" }
  ],
  type2 = [
    { type-name = "filterTransformView", filter = "bbox(geom,0,0,10,10)", transform = [ "dtg", "geom" ] }
  ]
}
```

Bulk Ingest into Accumulo

Database tables in Accumulo and HBase consistent of large, immutable files.

During normal operations, these are written by *compactions*.

When loading a large volume of data, compactions can slow down ingest.

GeoMesa 3.3 adds support for creating Accumulo RFiles via MapReduce jobs.

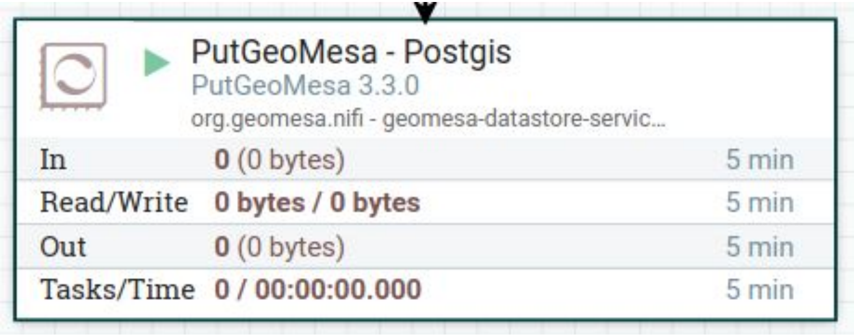
Previous versions of GeoMesa had such support for HBase.

This allows for building large tables more quickly (than using a traditional ingest).

NiFi: PostGIS Write Support

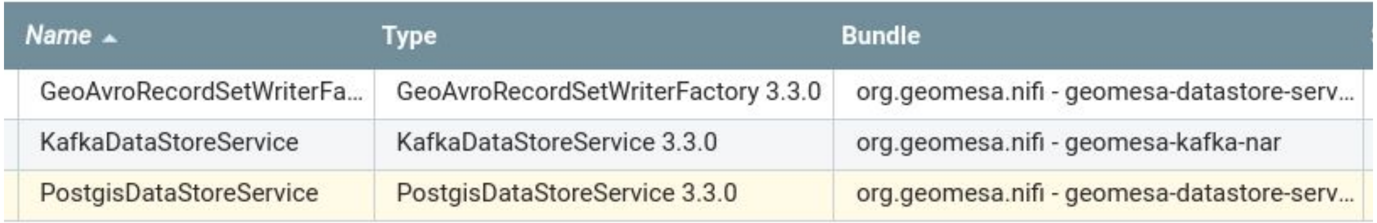
Data store connections have been moved out to NiFi controller services, simplifying NiFi configurations and allowing for re-use.

Postgis is now a supported data store controller service!



PutGeoMesa - Postgis
PutGeoMesa 3.3.0
org.geomesa.nifi - geomesa-datastore-servic...

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



Name ▲	Type	Bundle
GeoAvroRecordSetWriterFa...	GeoAvroRecordSetWriterFactory 3.3.0	org.geomesa.nifi - geomesa-datastore-serv...
KafkaDataStoreService	KafkaDataStoreService 3.3.0	org.geomesa.nifi - geomesa-kafka-nar
PostgisDataStoreService	PostgisDataStoreService 3.3.0	org.geomesa.nifi - geomesa-datastore-serv...

GeoMesa 3.4 / GeoMesa 4.0

TBD

Spark

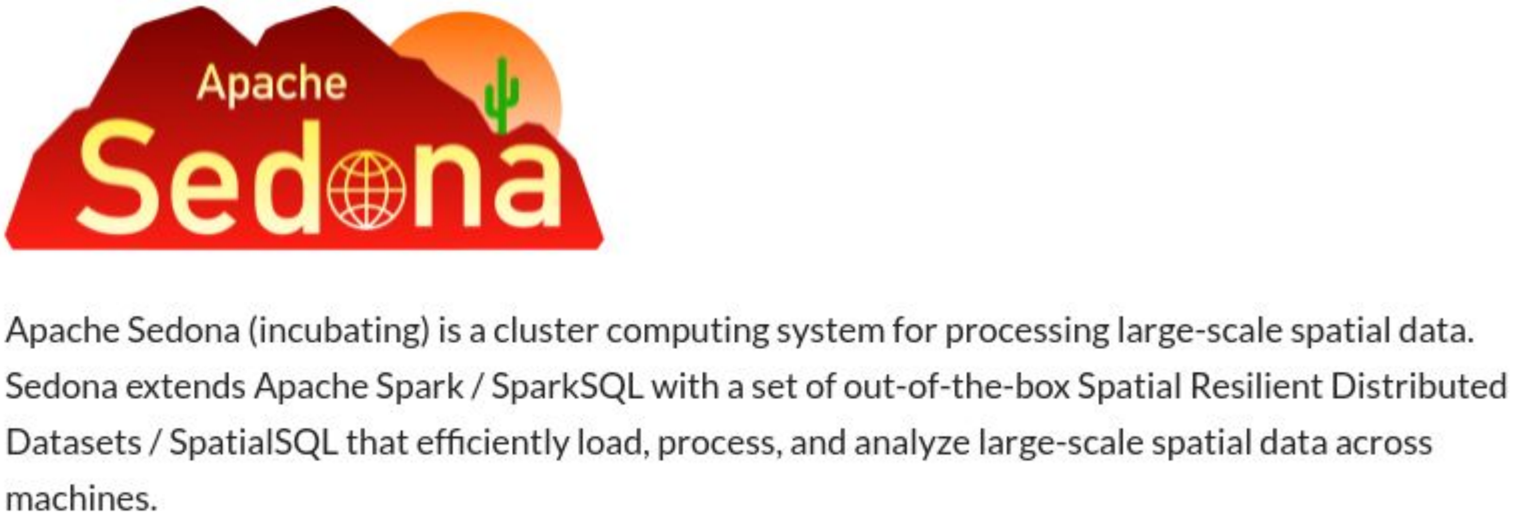
- Integration with Apache Sedona

Deprecations

Java 11 Support

WIP: Spark Integration with Apache Sedona

From https://sedona.apache.org/

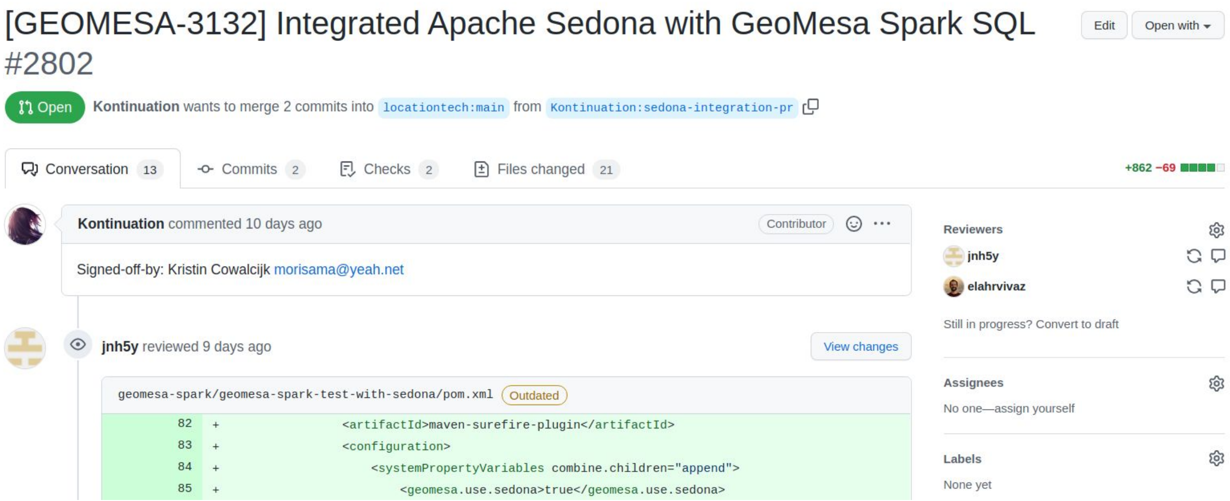


Apache Sedona (incubating) is a cluster computing system for processing large-scale spatial data. Sedona extends Apache Spark / SparkSQL with a set of out-of-the-box Spatial Resilient Distributed Datasets / SpatialSQL that efficiently load, process, and analyze large-scale spatial data across machines.

WIP: Spark Integration with Apache Sedona

Apache Sedona implements capabilities which optimize spatial joins in Spark.

There is a PR which provides a basic integration here: https://github.com/locationtech/geomesa/pull/2802



[GEOMESA-3132] Integrated Apache Sedona with GeoMesa Spark SQL #2802

Kontinuation wants to merge 2 commits into locationtech:main from Kontinuation:sedona-integration-pr

Kontinuation commented 10 days ago

Signed-off-by: Kristin Cowalcijk morisama@yeah.net

jnh5y reviewed 9 days ago

geomesa-spark/geomesa-spark-test-with-sedona/pom.xml

```
+                <artifactId>maven-surefire-plugin</artifactId>
+                <configuration>
+                    <systemPropertyVariables combine.children="append">
+                        <geomesa.use.sedona>true</geomesa.use.sedona>
```

Reviewers

jnh5y

elahrvivaz

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

None yet

Deprecations

The following modules have been deprecated and are slated for removal in 4.0:

- GeoMesa Kudu
- GeoMesa Streaming (Camel integration)
- GeoMesa Web
- GeoMesa GeoJSON

Java 11 Support

In order to support Java 11, GeoMesa will need to:

- Fix a few split packages
  - The team is waiting for the next major release to make this breaking change
- Test each of the backends and update accordingly
- Add support for cross building in CI/CD

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

GENERAL ATOMICS CCRi

Questions?

https://www.geomesa.org

https://gitter.im/locationtech/geomesa

jhughes@ccri.com

GENERAL ATOMICS
CCRi

Other Talks:

Introduction to Big Data Storage with LocationTech GeoMesa

2021-09-30, 15:30–16:00, Ushuaia

Streaming IoT sensor data with LocationTech GeoMesa, Apache Kafka, and NiFi

2021-10-01, 09:00–09:30, Bariloche