bakdata

End-to-end large messages processing with Kafka Streams & Kafka Connect

Philipp Schirmer | March 19th, 2020

Text-heavy data

Icons made by Pause08, Freepik, Zlatko Najdenovski, monkik from www.flaticon.com

Load data into Kafka, process and index in Elasticsearch

Amazon S3

Elasticsearch

https://aws.amazon.com/architecture/icons/
https://www.seekpng.com/ipng/u2q8r5r5w7e6q8t4_white-on-transparent-kafka-logo-svg/
https://cdn.worldvectorlogo.com/logos/elasticsearch.svg

Large messages in Kafka

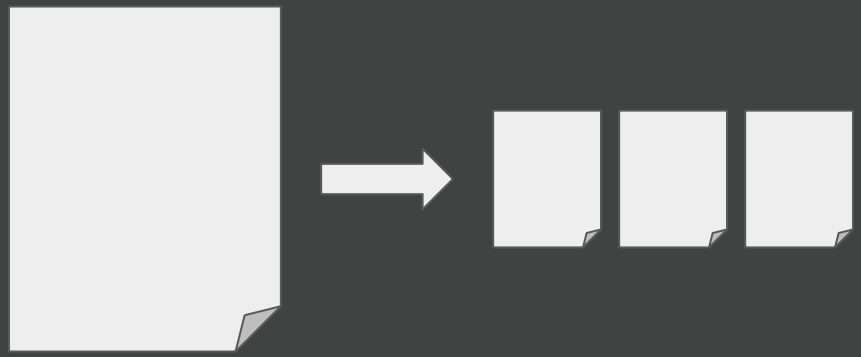
NLP-Operations require full document

But:

- Kafka prefers many small messages
- `max.message.bytes` defined by brokers (default 1MB)
- Limit cannot be configured in some scenarios (e.g., Confluent Cloud)
- There will always be (few) messages exceeding the limit

Solutions: Chunking

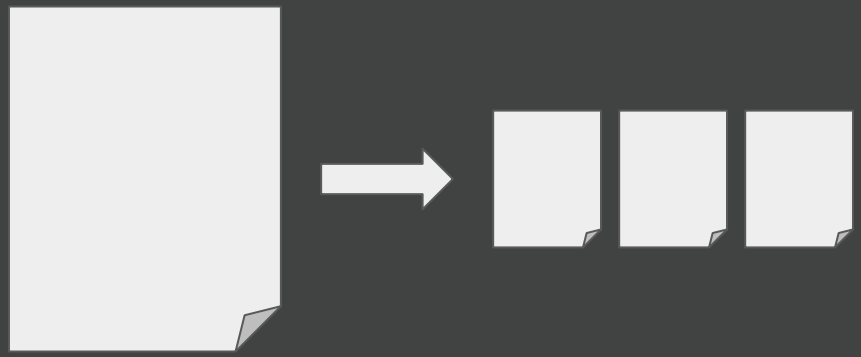
Split document into multiple small parts



bakdata

Solutions: Chunking

Split document into multiple small parts

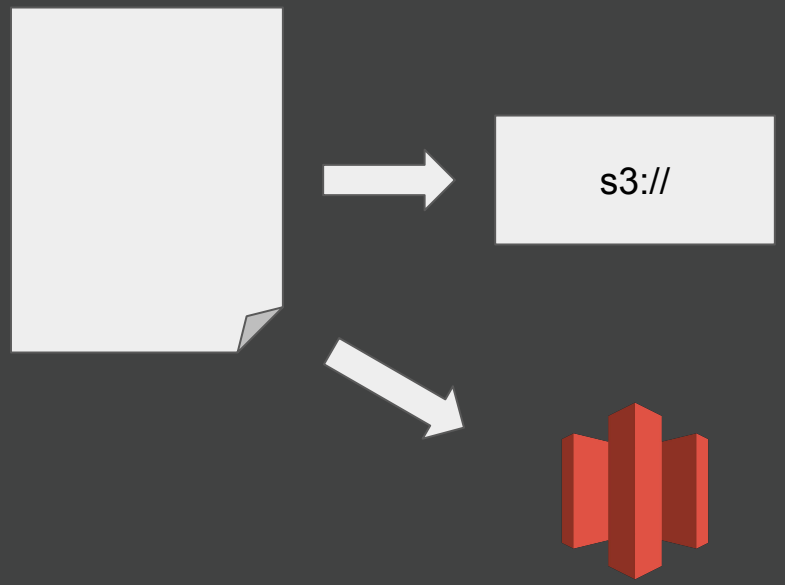


But: requires change of processing logic

- User must be aware of partial documents
- Expensive aggregations required

Solutions: External Storage

Store large messages in external system, e.g., Amazon S3, and send pointers to Kafka



bakdata

Kafka Streams SerDe

Kafka stores binary messages

SerDes offer an API to interpret raw messages with Kafka Streams (Java)

```
KStream<String, String> stream = builder.stream("my_topic", Consumed.with(Serdes.String(), Serdes.String()));
```


bakdata

S3-backed SerDe

Our S3-backed SerDe transparently handles storage and retrieval of messages

- Delegates actual (de-)serialization to a wrapped SerDe
- No changes to processing logic required, only configuration changes
- Small messages are sent to Kafka (almost) as before

Similar implementations for Kafka Connect and Faust (Python)

bakdata/kafka-s3-backed-serde (GitHub)
bakdata/faust-s3-backed-serializer (GitHub)

Serialization

1. Serialize message using actual SerDe
2. Check if message size in bytes exceeds configured limit
3.
    a. If message is small enough
        i. Send actual message small message flag to Kafka
    b. If message is too large
        i. Store message on S3
        ii. Send S3 URI and large message flag to Kafka

Serialization - S3 URI

S3 URIs must be unique

s3backed.base.path

key/value

s3://my-bucket/large-messages/my-topic/values/12ef79ab1097ebca1abe5982ce1763aa

topic

unique id

s3backed.id.generator

- Random UUID
- Murmur hash
- SHA-256

bakdata

Deserialization

1. Check flag
2.
    a. If flag denotes small message
        i. Proceed
    b. If denotes large message
        i. Retrieve message from S3
3. Deserialize message using actual SerDe

Demo

https://github.com/bakdata/s3-backed-serde-demo

Amazon S3

Elasticsearch

https://aws.amazon.com/architecture/icons/
https://www.seekpng.com/ipng/u2q8r5r5w7e6q8t4_white-on-transparent-kafka-logo-svg/
https://cdn.worldvectorlogo.com/logos/elasticsearch.svg

tf-idf

… is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

$$\mathrm{tf}(t, D) = \frac{\#(t, D)}{\max_{t' \in D} \#(t', D)}$$

$$\mathrm{idf}(t) = \log \frac{N}{\sum_{D: t \in D} 1}$$

$$\mathrm{tf.idf}(t, D) = \mathrm{tf}(t, D) \cdot \mathrm{idf}(t)$$

https://en.wikipedia.org/wiki/Tf%E2%80%93idf
https://de.wikipedia.org/wiki/Tf-idf-Ma%C3%9F

bakdata

S3-backed SerDe - Retention

Kafka topics have a message retention

S3 supports object expiration

- Each rule applies to a prefix → each topic has its own prefix
- Timestamp of S3 object and Kafka message can differ

S3-backed SerDe - Limitations

- Communication with S3 is costly
- Small message overhead (1 byte)
- Partitioning affected if used for keys
- Dangling references if compaction is used
- No KSQL support (KSQL Formats)

Read more

[Processing Large Messages with Kafka Streams (Medium)]

bakdata ABOUT | WWW.BAKDATA.COM

Processing Large Messages with Kafka Streams

Philipp Schirmer
Feb 20 · 5 min read ★

bakdata

Read more

[bakdata/kafka-s3-backed-serde (GitHub)]

Azure Pipelines succeeded | quality gate passed | coverage 94.6% | maven-central v1.1.2

kafka-s3-backed-serde

A Kafka Serde that reads and writes records from and to S3 transparently.

Getting Started

Serde

You can add kafka-s3-backed-serde via Maven Central.

Gradle

```
compile group: 'com.bakdata.kafka', name: 's3-backed-serde', version: '1.1.0'
```

Maven

bakdata

Read more

[bakdata/faust-s3-backed-serializer (GitHub)]

license MIT | python 3.6 | 3.7 | 3.8 | Azure Pipelines succeeded | pypi package 1.0.0

faust-s3-backed-serializer

A Faust Serializer that reads and writes records from and to S3 transparently.

This serializer is compatible with our Kafka S3-backed SerDe for Java.

Read more about it on our blog.

Getting Started

PyPi

```
pip install faust-s3-backed-serializer
```

Usage

bakdata

Contact us

community@bakdata.com

@bakdata

https://www.hiclipart.com/free-transparent-background-png-clipart-qbavj
https://upload.wikimedia.org/wikipedia/en/thumb/9/9f/Twitter_bird_logo_2012.svg/300px-Twitter_bird_logo_2012.svg.png