swissbib

Apache Flink
a platform for distributed Data (stream) Processing and Data Analytics

elag 2019, lightning talk
Berlin, 10.5.2019

Günter Hipler – Systemarchitekt, swissbib

Apache Flink – a few key figures

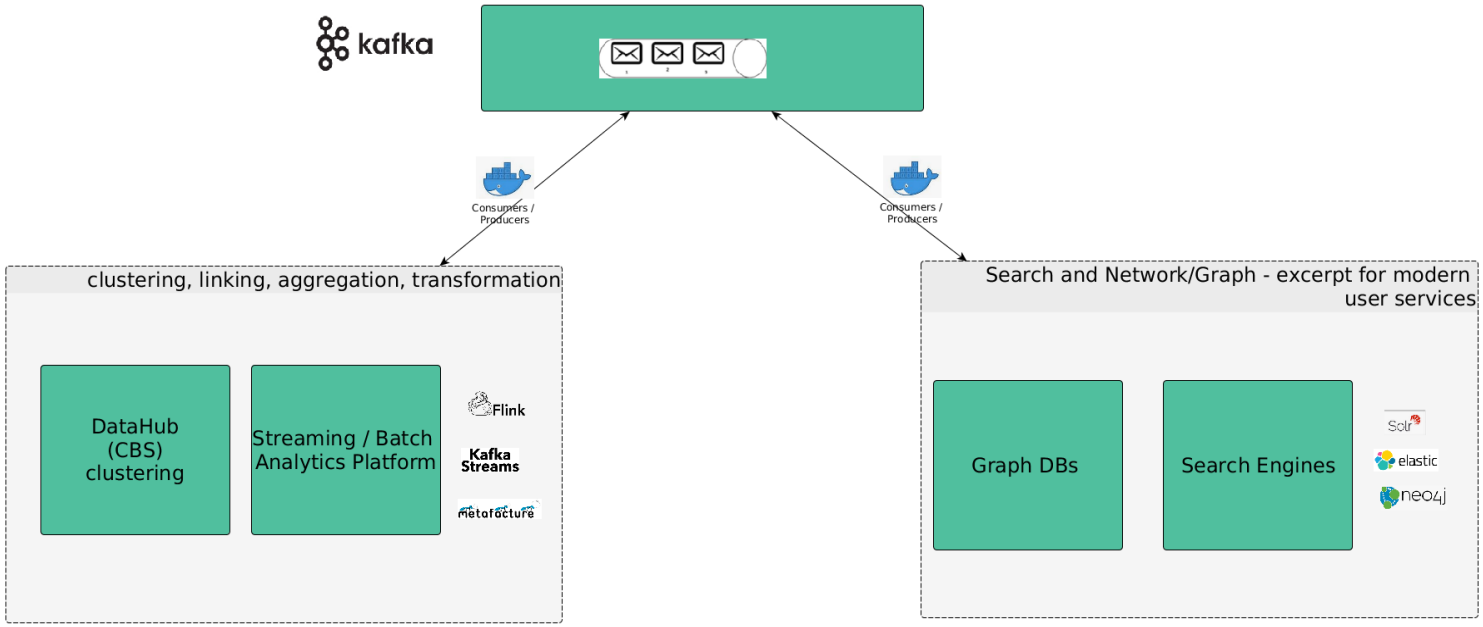
- Apache top level project since 2014
- Originally developed at European universities (Berlin, Potsdam)
- Until end of 2018 mainly developed by Berlin based company ververica.com
- January 2019, ververica was bought by «Chinese Amazon» Alibaba (~100 Million Euros)

I argue that: (for discussion at Elag)

If scientific libraries want to remain visible providers of digital information as a basis for user services even in 10 years' time, they must use the tools and procedures which until now have mostly only been used by large commercial companies – and to combine them with our own domain specific possibilities.

This is one important pillar to retain data sovereignty.

current Flink/swissbib Use Case (Transformation pipeline CBS - Marc - SolrDocs)



available library domain specific data tools (CBS, metafacture, ...)
are going to be combined with modern data-processing
and data-analytics software (Kafka, Flink, ...)

Code example 1: setup of Flink cluster

```
public class DocProcEngine {

    public static void main (String[] args) throws Exception {

        //setup Flink cluster environment
        final StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
        Properties properties = new Properties();
        properties.setProperty("bootstrap.servers", "kafka1:9092,kafka2:9092,kafka3:9092");
        properties.setProperty("group.id", "test4");

        //setup Flink Kafkaconsumer (Marc data coming from CBS)
        FlinkKafkaConsumer<FlinkSbMetadaModel> fc = new FlinkKafkaConsumer<~>(     topic: "sb-all",
                                                    new KeyedSwissbibFlinkMetadataSchema(),
                                                    properties);
        fc.setStartFromEarliest();

        //setup Kafka producer to serialize (transformed Solr-Docs)
        FlinkKafkaProducer<FlinkSbMetadaModel> kafkaProducer = new FlinkKafkaProducer<~>( topicId: "sb-solr",
                                                    new KeyedSwissbibFlinkMetadataSchema() ,
                                                    properties);

        DataStream<FlinkSbMetadaModel> stream = env.addSource(fc); //add Kafka data source
        stream.map(new DocProcFunction()) // add map - transformation Function
            .addSink(kafkaProducer);
        env.setParallelism(1); //number of parallel tasks on the Flink cluster
        env.execute( jobName: "swissbib classic goes BigData");
    }
}
```

https://github.com/swissbib/content2SearchDocs/blob/integrateMF.v2/src/main/java/org/swissbib/flink/DocProcEngine.java

Code example 2: Flink map function
Marc – Solrdoc (with integrated Metafacture pipeline)

```java
public class DocProcFunction extends RichMapFunction<FlinkSbMetadaModel,FlinkSbMetadaModel> {

    MFXsltBasedBridge bridge2pipe;

    @Override
    public FlinkSbMetadaModel map(FlinkSbMetadaModel record) throws Exception {
        //the map (as part of the Flink data-flow graph)
        // is called for each Marc-Record (event in Flink language)

        //each event is sent into the Metafacture - pipe and transformed
        //into a Solr document
        String response = bridge2pipe.transform(record.getData());

        //the transformed event is prepared for the next Flink operator (Kafka-sink)
        FlinkSbMetadaModel sbm = new FlinkSbMetadaModel();
        sbm.setData(response);
        sbm.setCbsAction(CbsActions.CREATE);
        return sbm;

    }

    @Override
    public void open(Configuration parameters) throws Exception {
        super.open(parameters);
        //at the beginning initialization of the Flink map function
        //here create the Metafacture transformation pipe via configuration
        // not flux - just for Metafacture experts!
        bridge2pipe = new MFXsltBasedBridge( configFileName: "pipeDefaultConfig.yaml");
        bridge2pipe.init();
    }
```

https://github.com/swissbib/content2SearchDocs/blob/integrateMF.v2/src/main/java/org/swissbib/flink/DocProcFunction.java

Thanks!

Disclosure:
We are currently in the phase of setting up our new data processing and data analytics platform.
We learn new things every day and the code presented in this short demo will be definitely matter of change.

Günter Hipler

more www.swissbib.ch resources:
https://github.com/swissbib https://gitlab.com/swissbib
https://twitter.com/swissbib https://swissbib.blogspot.com/