CERN

Pilot Kafka Service

Manuel Martín Márquez

Kafka

- Kafka is a distributed streaming platform
  - High Scalable (partition)
  - Fault Tolerant (replication)
  - Allow high level of parallelism and decoupling between data producers and data consumers
- De facto standard for near real-time store, access and process data streams
- Critical component of most of the Big Data Platform and therefore of Hadoop ecosystem

Kafka Basic Concepts

Broker: Kafka node on the cluster

Topics: Stream of records category

- Multiple writers and readers
- Partitioned
- Replicated

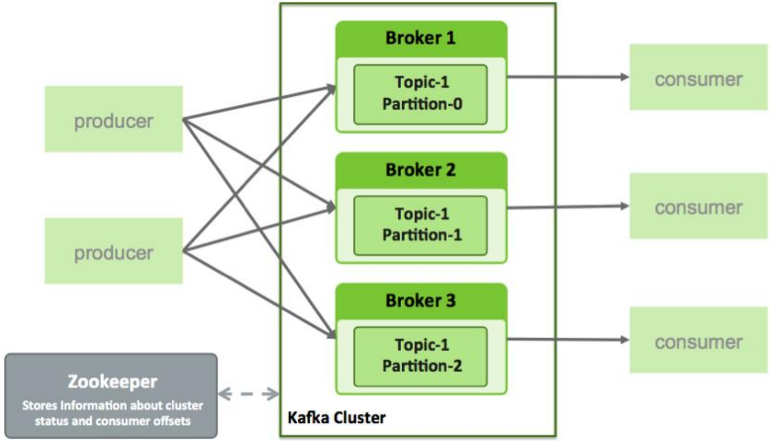
Consumer: pulls messages off of a Kafka topic

Producer: push messages into a Kafka topic

Data Retention:

- Based on time or size

Zookeeper: Stores Kafka Metadata

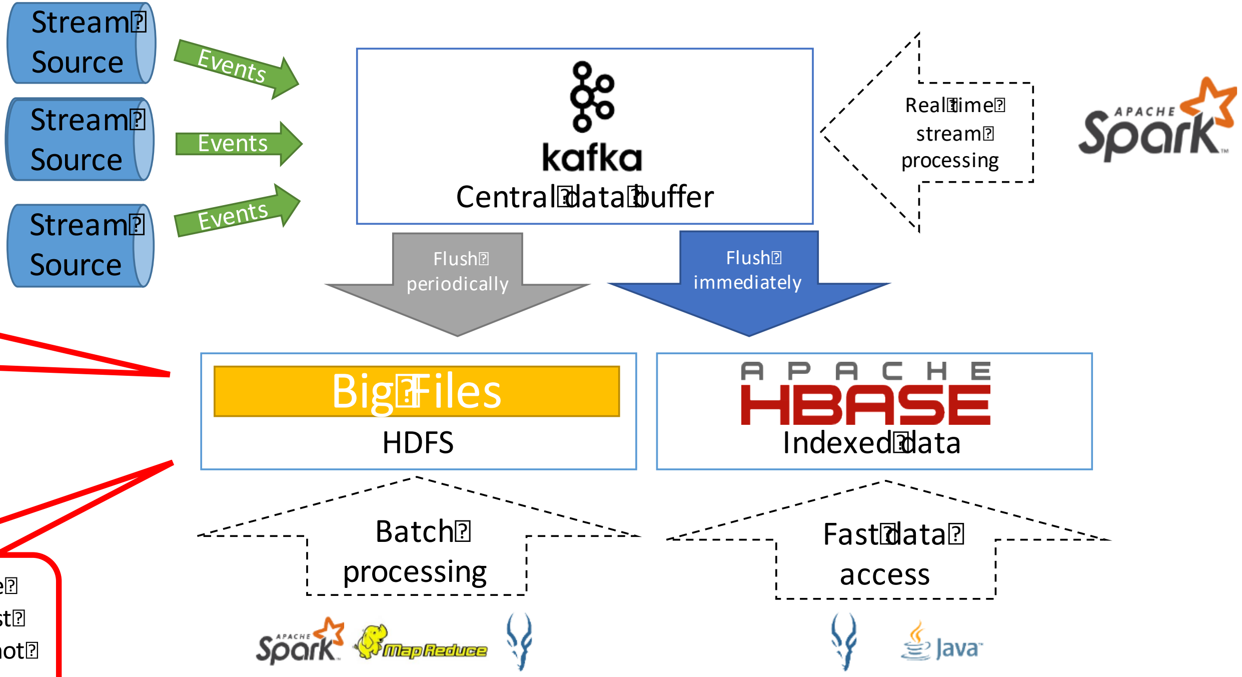


Source: Hortonworks

Kafka entry points

- Custom implementation of producer and consumer using Kafka client API
  - Java, Scala, C++, Python
- Kafka Connectors
  - LogFile, HDFS, JDBC, ElasticSearch…
- Logstash
  - Source and sink
- Apache Flume out-of-the-box can use Kafka as
  - Source, Channel, Sink
- Other ingestion or processing tools support Kafka
  - Apache Spark, LinkedIn Gobblin, Apache Storm…

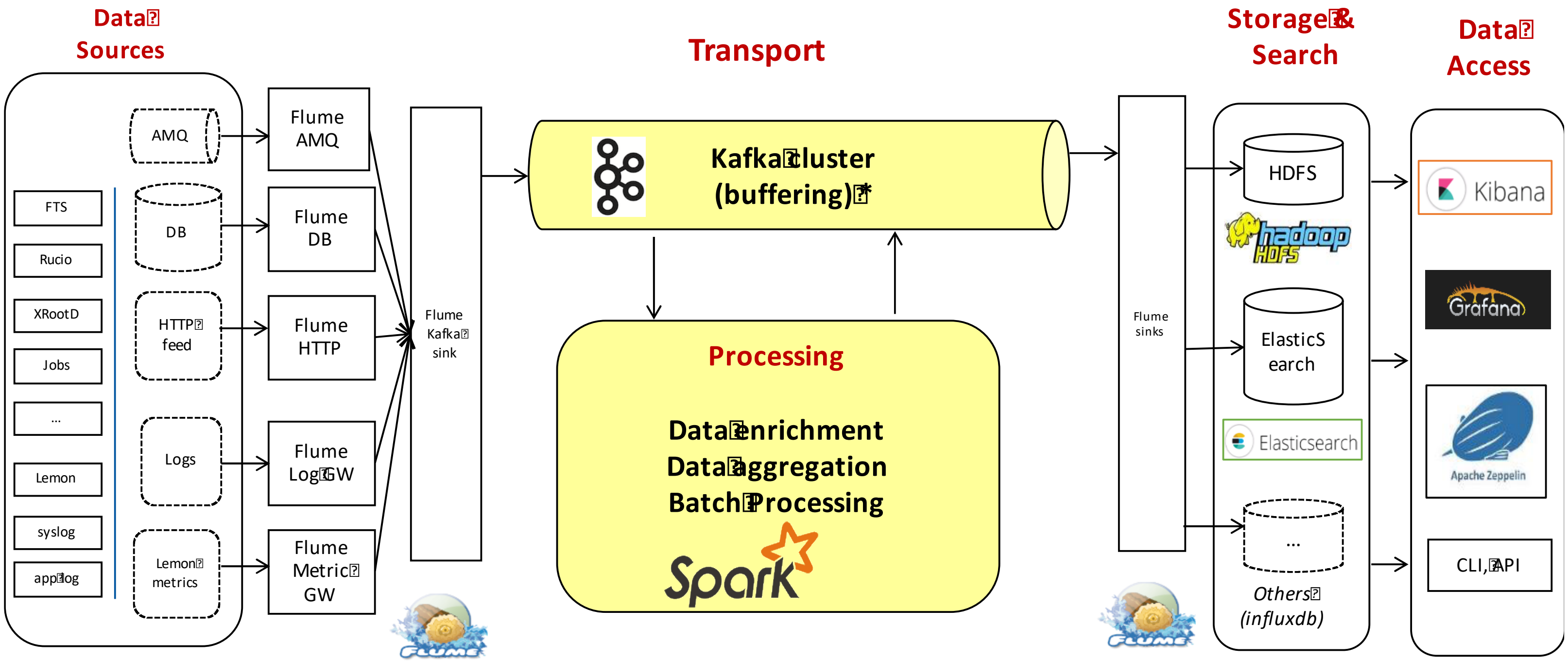
Kafka for Data Integration and Processing



No data lost during downtime (scheduled and unscheduled) of a Hadoop cluster

Kafka buffers protects the recent data from being lost (before a daily HDFS snapshot can backed them up)

Kafka at CERN – it monitoring



Data Sources

FTS

Rucio

XRootD

Jobs

...

Lemon

syslog

app log

AMQ

DB

HTTP feed

Logs

Lemon metrics

Flume AMQ

Flume DB

Flume HTTP

Flume Log GW

Flume Metric GW

Flume Kafka sink

Transport

Kafka cluster (buffering) *

Processing

Data enrichment
Data aggregation
Batch Processing

Spark

Flume sinks

Storage & Search

HDFS

hadoop HDFS

ElasticSearch

Elasticsearch

...

Others (influxdb)

Data Access

Kibana

Grafana

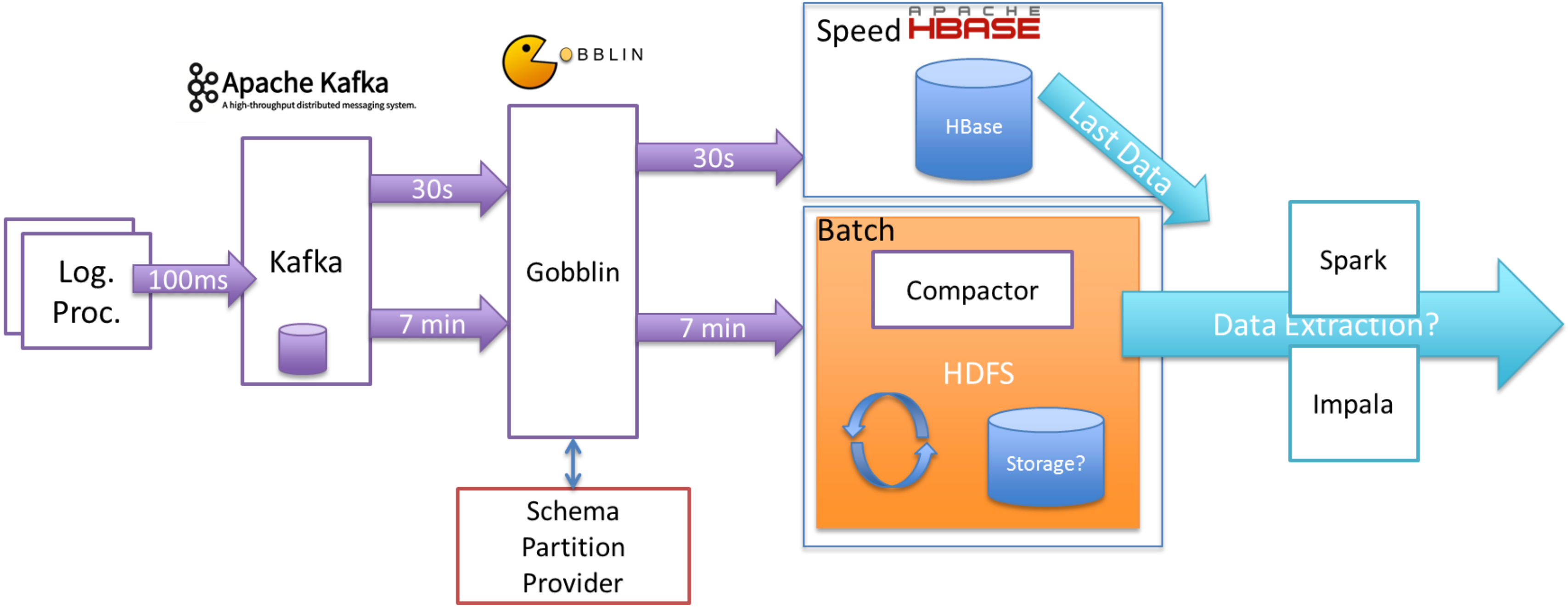
Apache Zeppelin

CLI, API

Kafka at CERN – it monitoring (Requirements)

- Throughput and retention policy
  - Currently 200 GB/day (forecast 500 GB/day)
  - Retention Policy 12h in qa and 24 hours in prod (largest retention policy to cover potential problems over weekends)
  - ~ 4000 messages, up to 10k peaks
  - ~50 topics
- Security (Kerberos)
  - Flume can be potentially upgrade to 1.7 early in 2017 (work in progress already).
- Administration Capabilities
  - Administrative operations
  - Topic configuration, rebalancing, user management, start/stop cluster
  - Possibility to increase retention policy, replication factor

Kafka at CERN – CALS



Kafka at CERN – CALS (Requirements)

- Throughput and retention policy
  - Currently 30 GB/hour only including the logging processes
  - Plan to incrementally include all the systems with potentially mean several TBs
  - Compression with Snappy will be evaluated to determined performance
  - Retention policy 24 hours, which is the time they need to buffer data and compact it to send it to Hadoop
- Security (Kerberos)
- Infrastructure
  - Openstack under several conditions:
    - TN need to be supported for several reasons
    - High availability of the service CALS on top of private cloud (No CALS no BEAM in the LHC)
- Administration Capabilities
  - Administrative operations
  - Topic configuration, rebalancing, user management, start/stop cluster
  - Possibility to increase retention policy, replication factor

Kafka at CERN

- Security Team
  - Already using Kafka for pattern matching
  - Data integration
- LHC Postmortem
  - Potentially ingested by CALS
- Industrial Control Systems
  - WinCCOA Data

Pilot Kafka Service

- Scope
  - Study the current Kafka use case together with the different teams involved
  - Collect requirements
  - Understand feasibility and added value of Kafka as a central service

Pilot Kafka Service

- Collect requirements (5 Major Use Cases):
  - **CALS, IT-Monitoring**, Security Team, Industrial Control, Post-mortem
  - Throughput, Retention Policy, Security, Infrastructure, Administration Capabilities
  - Agreement to test the service from the first phase
    - Ensure the service cope with their requirements
  - More details: https://twiki.cern.ch/twiki/bin/viewauth/DB/CERNonly/KafkaService

Pilot Kafka Service – Current Development

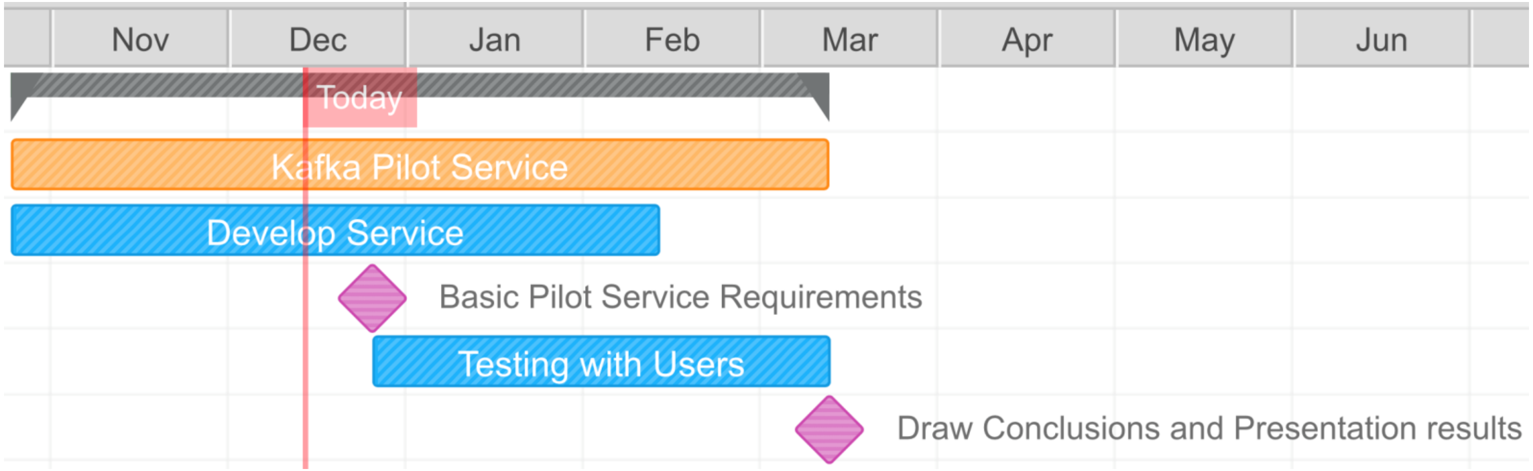
- Pilot Implementation - rapid iteration which will help to understand service and use case.
  - On-demand Kafka service approach
    - Self-Service Cluster creation, management and expansion
    - Allow users to perform administrative tasks that are traditionally carried out by administrators
    - Facilitating operating system and engine updates (Kafka, Zookeeper)
    - Transparently integrate all the needed services (Security, Storage, Procurement, etc)
    - Support for service continuity in case of hardware failure

Pilot Kafka Service – Current Development

- Configuration and Management REST API
- Security enabled - Kerberos on Kafka and Zookeeper (SSL optional)
- Monitoring Capabilities
- OpenStack on GPN
- Network storage
- Dedicated Kafka and Zookeeper per user

Towards Kafka Production Service

- Service evaluation phase and time line



Towards Kafka Production Service

- Consolidation to Production
  - Web Interface to manage clusters (Self-service)
  - Evolution of the configuration management API
    - Functionalities toward the self-service platform
    - Integration with Openstack
  - Full monitoring beyond JMX metrics
  - Kafka-Mirroring (High Availability)
  - Deploy service in TN (due to service design that is transparent for us)
    - Kafka as close as possible to consumers and producers