Data Integration with Apache Kafka

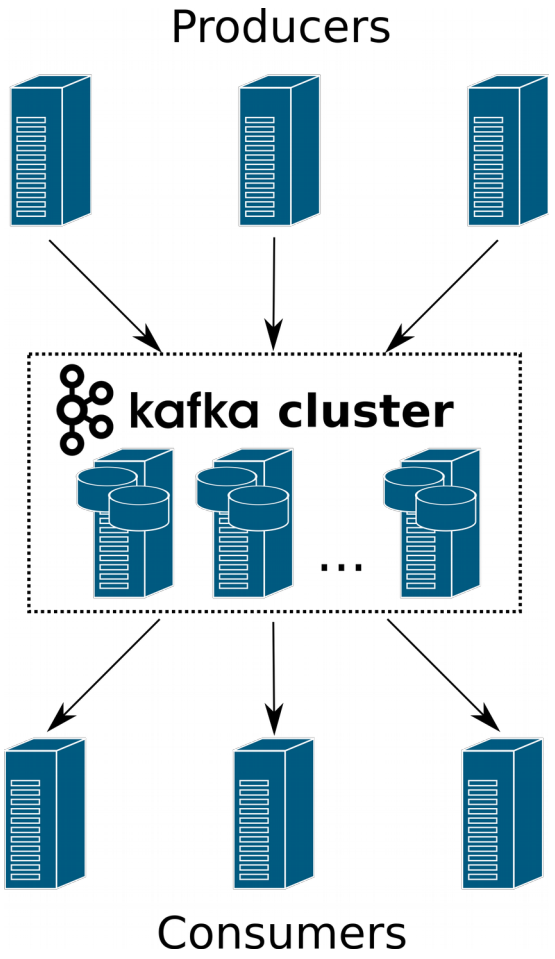
- Quick reminder of Kafka basics + terminology
- Kafka Connect
  - What it is
  - What Connectors are available
- Stream Processing
  - What it is
  - Example technologies
  - Focus on C++ technologies
- Demo

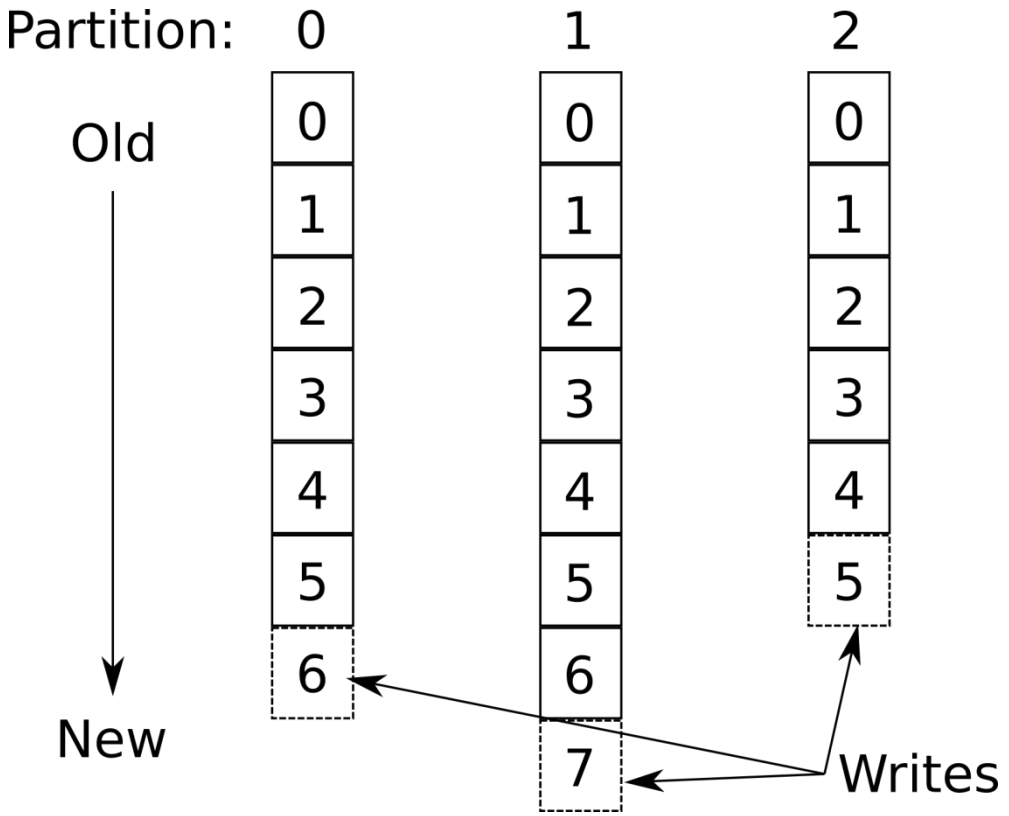
Data Integration with Apache Kafka

- **Quick reminder of Kafka basics + terminology**
- Kafka Connect
  - What it is
  - What Connectors are available
- Stream Processing
  - What it is
  - Example technologies
  - Focus on C++ technologies
- Demo

Apache Kafka

A high-throughput distributed messaging system.



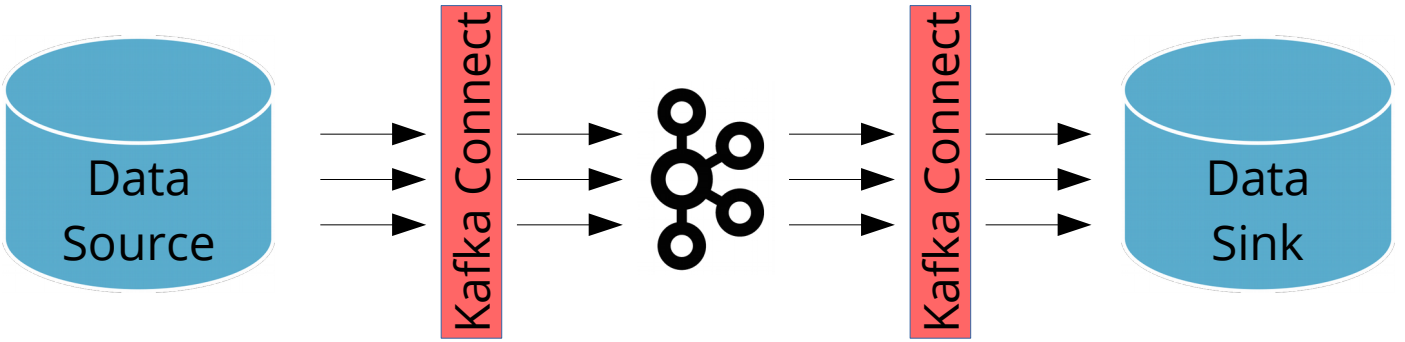

Partition:
0
1
2
Old
0
1
2
3
4
5
6
0
1
2
3
4
5
6
7
0
1
2
3
4
5
New
Writes

Data Integration with Apache Kafka

- Quick reminder of Kafka basics + terminology
- **Kafka Connect**
  - What it is
  - What Connectors are available
- Stream Processing
  - What it is
  - Example technologies
  - Focus on C++ technologies
- Demo

Kafka Connect

- Kafka Connect is an API for writing source and sink *Connectors*



- Scalable, across processes (1 per partition) and machines
- 10s/100s already implemented
- Supports Avro and JSON serialisation

A few examples:

Storage

elasticsearch

Logging

Messaging

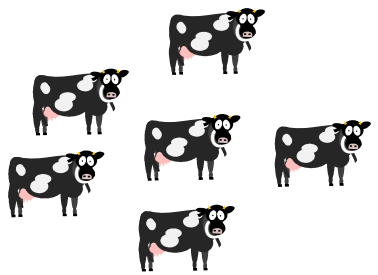
#irc

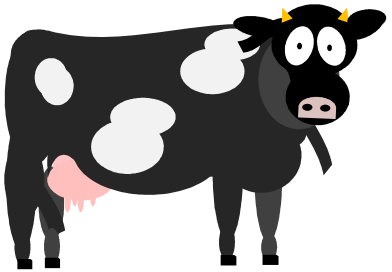
Processing

Data Integration with Apache Kafka

- Quick reminder of Kafka basics + terminology
- Kafka Connect
  - What it is
  - What Connectors are available
- **Stream Processing**
  - What it is
  - Example technologies
  - Focus on C++ technologies
- Demo

Stream Processing





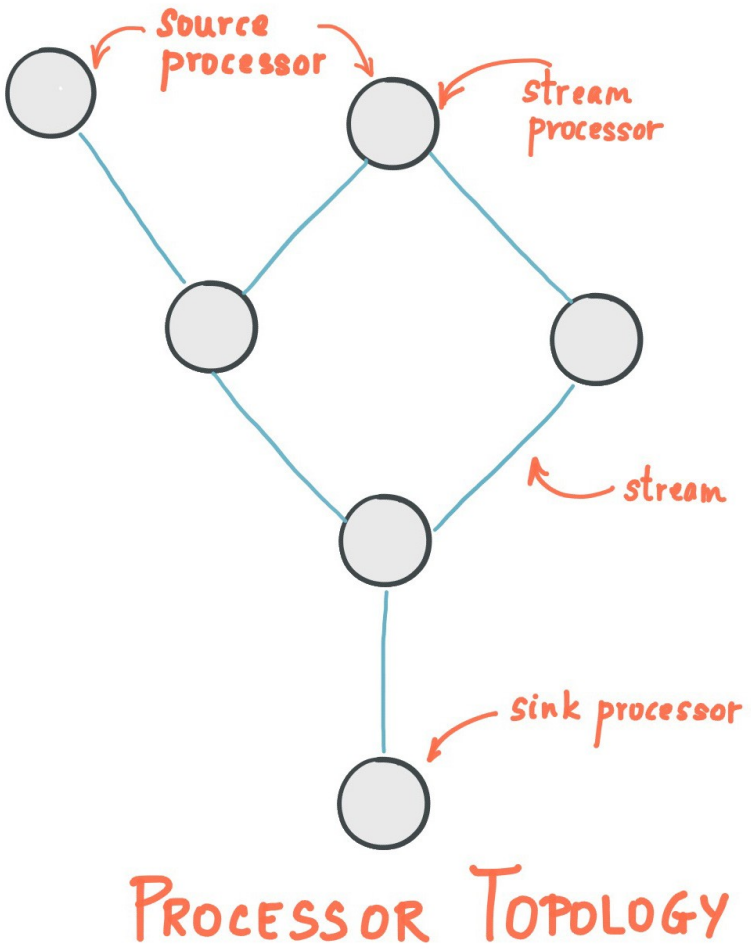
Stream

Batch

Unbounded
Frequent small data

Bounded
Occasional large data

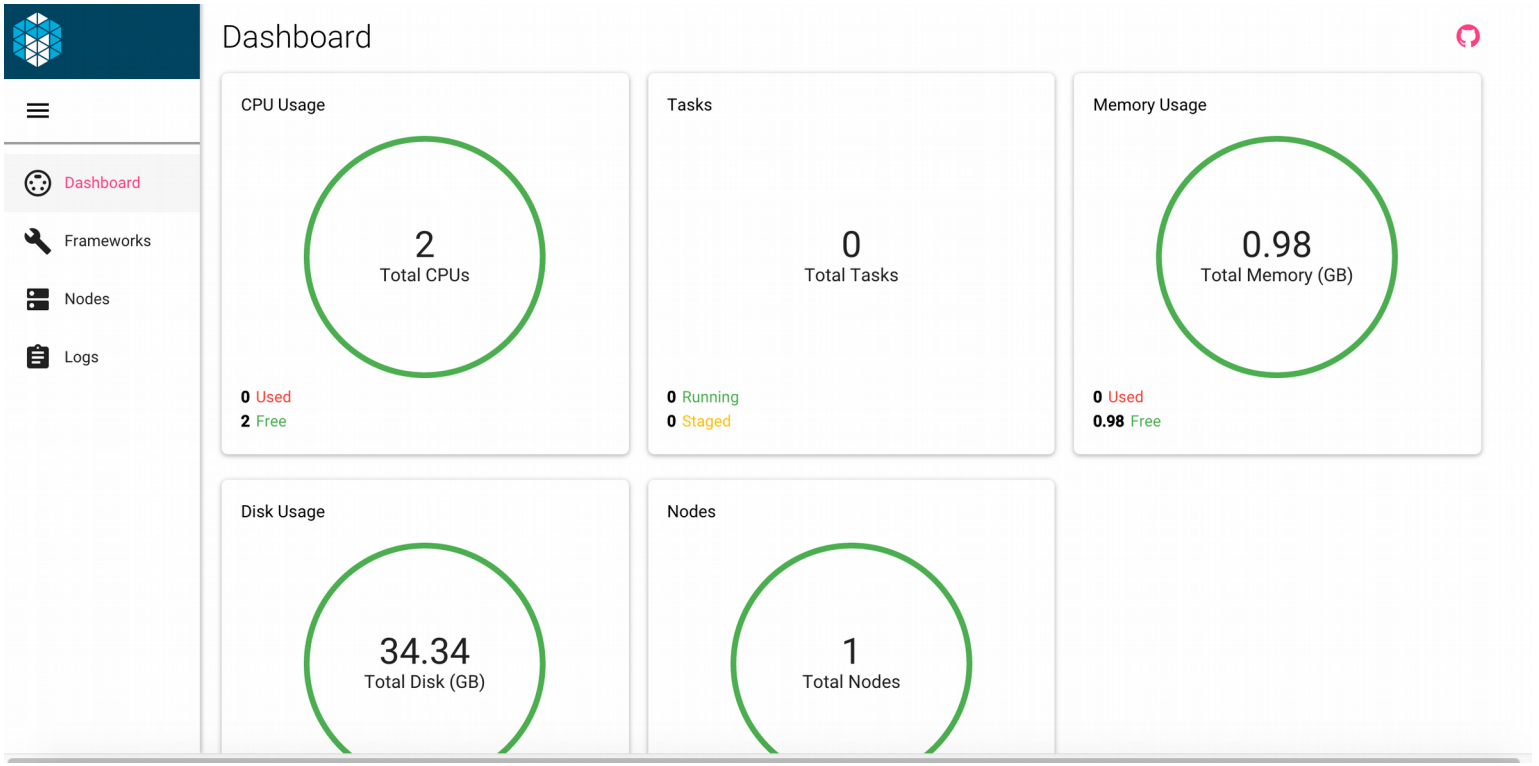
- Real-time processing, left side + time constraint



- **Compositional** - explicitly code topology with sources, sinks etc **OR**
- **Declarative** - high level functional code, system creates and optimises the topology itself

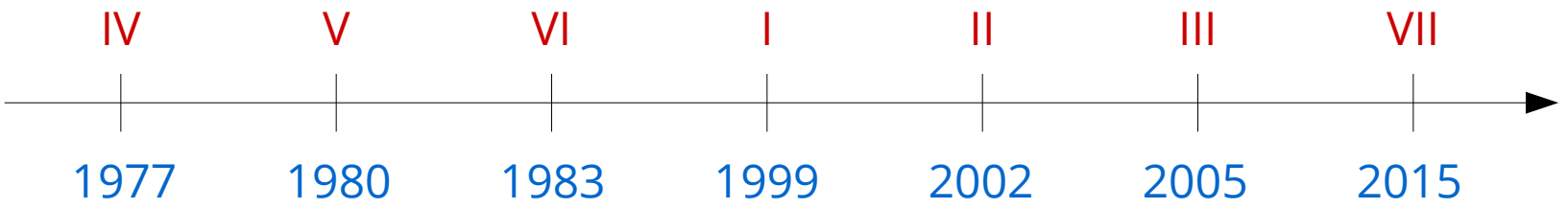
Resource Management

- Use processing cluster machines efficiently
- Deployment
- Monitoring



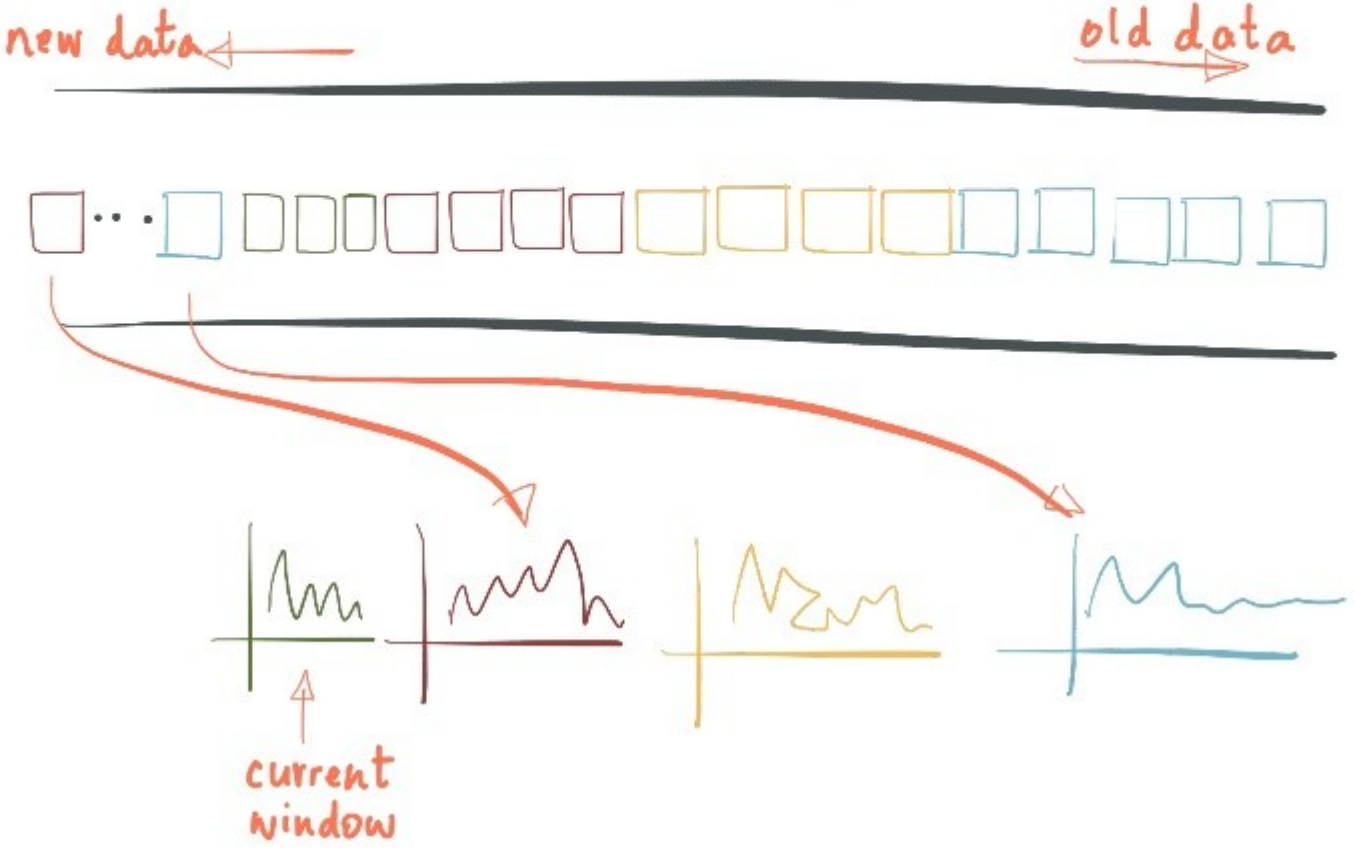
Out of Order Events

Event Time



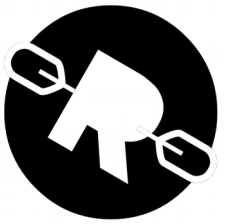
Processing Time

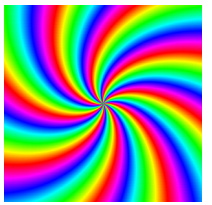
WINDOWING



Technologies

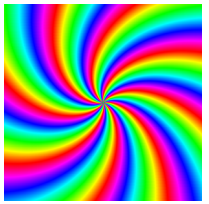
- Apache processing technologies:
  - Flink, Spark, Storm, Samza etc
  - All in JVM (Java, Scala, Clojure), some have Python bindings
- C++:
  - RaftLib, Thrill, *et al*
  - Concord
  - Kafka Streams (soon)





Non-Centralised Stream Processing

- RaftLib (on cppcast recently) and Thrill
- Modern C++
- Support heterogeneous hardware (CUDA etc)
- Beta versions, incomplete, maybe unstable
- No centralised broker for communications
  - Monitoring difficult
  - lack of fault tolerance



Kafka Streams

- Lightweight stream processing API with:
  - Windowing with out-of-order handling
  - Spans full range from stream to batch
  - Monitoring – Kafka-manager, jmx metrics
- New but already widely used and well documented due to Kafka popularity
- Not implemented yet in C++ library, on roadmap

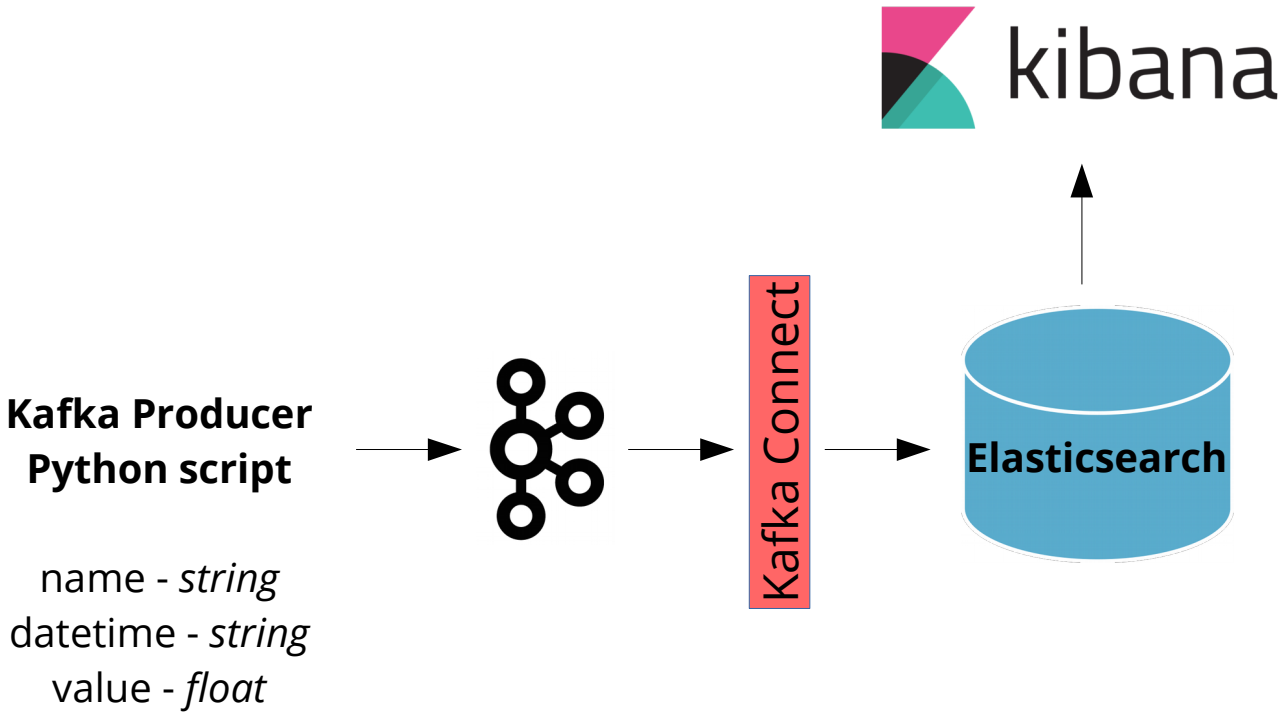
- A rare complete C++ option
- Initial impression – Retrofitting Mantid may be difficult?
- But, tech it is built on might be useful…

- Mesos – distributed systems kernel with C++ API
- Deploy container, MPI rank or whatever on a machine in the cluster which has spare resources
- Web-UI for monitoring

DEPLOYMENT COOLNESS SPECTRUM

Data Integration with Apache Kafka

- Quick reminder of Kafka basics + terminology
- Kafka Connect
  - What it is
  - What Connectors are available
- Stream Processing
  - What it is
  - Example technologies
  - Focus on C++ technologies
- **Demo**


kibana
Kafka Producer
Python script
name - string
datetime - string
value - float
Kafka Connect
Elasticsearch

Summary

- Kafka Connect provides means to get data between Kafka and many other technologies
  - Elasticsearch Connector – for Kibana/Grafana
- Most stream processing technologies probably not useful to us, but
  - Time windowing – Kafka Streams
  - Maybe Concord
  - Resource management - Mesos