Data LIBERATION and data INTEGRATION with Apache Kafka

Martin Kleppmann @martinkl
Designing Data-Intensive Applications

Martin Kleppmann

dataintensive.net

@martinkl
Maslow's Hierarchy of Needs

- self-actualization
- esteem
- love, belonging
- safety
- food & water
DATA NEEDS

vision/mission
DATA NEEDS

vision/mission

products

data science
DATA FRAGMENTATION

- Relational DBs
- NoSQL DBs
- Log files
- Message queue
- Search indexes
- Monitoring
EXTRACT

TRANSFORM

LOAD
1990 Data warehousing
1990  Data warehousing

2008  Hadoop & MapReduce
1990

Data warehousing
- Drop relational assumption
- Programmability
- Open Source

2008

Hadoop & MapReduce
1990
Data warehousing
- Drop relational assumption
- Programmability
- Open Source

2008
Hadoop & MapReduce

2015
Kafka & streaming data
1990

Data warehousing
- Drop relational assumption
- Programmability
- Open Source

2008

Hadoop & MapReduce
- Batch $\Rightarrow$ real-time
- Daily $\Rightarrow$ continuous

2015

Kafka & streaming data
Kafka

APACHE SOFTWARE FOUNDATION
Event streams
“something happened”

Event

stream

subscribe to it
stream

oldest events

most recent events
stream

new events added here

← oldest events

most recent events →
stream

new events added here

← oldest events

real-time consumer position (close to head of stream)

most recent events →
"SOMETHING HAPPENED"

- User x clicked link y (activity event)
"SOMETHING HAPPENED"

- User x clicked link y
  (activity event)

- Sensor x sent reading y
  (time series tick)
"SOMETHING HAPPENED"

- User x clicked link y (activity event)
- Sensor x sent reading y (time series tick)
- Database record x updated to y (data change event)

Immutable, point in time
DATA STREAMS - FRESHLY BOTTLED AT SOURCE
A user interacts with the Web app, which reads and writes data to Postgres. A BW plugin is used to get snapshot and extract schemas, and to change streams. A Bottled Water client daemon is also involved.
Web app → Postgres

Web app reads and writes to Postgres as usual.

Postgres has a Bottled Water (BW) plugin.

Bottled Water client daemon:
- Gets snapshot
- Extracts schemas
- Change stream
- Publish snapshot & changes

Kafka
Web app → Postgres
(read/write)
(as usual)

get snapshot & extract schemas
change stream,

Bottled Water client daemon

Kafka

Consumers

publish snapshot & changes
Web app \[\rightarrow\] Postgres

Avro schema registry \[\leftarrow\] register schemas

Bottled Water client daemon

Kafka

Consumers

get snapshot & extract schemas
change stream,
publish snapshot & changes
Data access

Make all data available as streams, including DBs
1. **Data access**
   Make all data available as streams, including DBs

2. **Common data format**
   Metadata: Schema, semantics, provenance, evolution
Local optimum

Using your favorite data format
Local optimum

Using your favorite data format

Global optimum

Standardizing on one format
JSON

+ widely supported
+ human-readable (ish)
JSON

+ widely supported
+ human-readable (ish)

- integer/floating-point mess
**JSON**

+ widely supported
+ human-readable (ish)
- integer/floating-point mess
- no binary strings
JSON

+ widely supported
+ human-readable (ish)

- integer/floating-point mess
- no binary strings
- verbose & slow
record PageViewEvent {
    long timestamp;
    string pageURL;
    union { IPv4Addr, IPv6Addr }
        clientIP;
}
record PageViewEvent {
    /** milliseconds since epoch */
    long timestamp;
    /** path and query params */
    string pageURL;
    /** IP address of client */
    union { IPv4Addr, IPv6Addr } clientIP;
    ...
}
record PageViewEvent {
    long timestamp;
    string pageURL;
    union { IPv4Addr, IPv6Addr } clientIP;
    union { null, string } sessionID = null;
}
Record `PageView Event`:

```java
long timestamp;
string pageURL;
union IPv4Addr, IPv6Addr? clientIP;
```
Record `PageView Event`:

```java
/** milliseconds since epoch */
long timestamp;

/** path and query params */
string pageURL;

/** client IP address */
union {[IPv4Addr, IPv6Addr] clientIP;

```
Record PageView Event

/** milliseconds since epoch */
long timestamp;

/** path and query params */
string pageURL;

/** client IP address */
union [IP4Addr, IP6Addr] clientIP;

timestamp
pageURL
clientIP

Message encoding
Backward compatibility

Producers

Schema registry

Consumers

v1

v1

v1

v2

v2

v2
Mixed versions

Producers

v23
v18
v21

Schema registry

Consumers

v19
v12
v22
Schema registry

Version number assignment

Compatibility check

Documentation
1. Data access
   Make all data available as streams, including DBs

2. Common data format
   Metadata: Schema, semantics, provenance, evolution
1. Data access
Make all data available as streams, including DBs

2. Common data format
Metadata: Schema, semantics, provenance, evolution

3. Composability
Loosely coupled stream processors, Kafka as pipe
Ad impressions \rightarrow \text{INVENTORY} \rightarrow \text{JOIN} \rightarrow \text{Impr + Clicks} \rightarrow \text{AGG} \rightarrow \text{RELEVANCE} \rightarrow \text{BILLING} \rightarrow \text{Charges} \rightarrow \text{PAYMENT}
Central hub for event streams
Central hub for event streams
Central hub for event streams
Central hub for event streams

Team A

Team B

Team C
Central hub for event streams

Team A

Team B

Team C

Team D
References


Office hours:

5.25pm today

O’Reilly Booth
Expo Hall
Discount code: **TS2015**
50% off ebooks

---

**FREE SIGNED COPIES!**

**Thu 3.35pm**

@O'Reilly Booth

Expo Hall